

SyncWare News

The journal for technical
applications of T/S
computers

Volume 4 Number 2

Nov.-Dec. '86

Season's Greetings

It's hard to believe that another Holiday Season is upon us again. Soon we will be into a new year. Tom Woods, Fred Nachbaur, Tom Bent, and myself would like to offer you our best wishes for the Holidays and the coming New Year. - Jeff Moore

Attention Programmers!!

The Second Sort-of-Annual Programmers Competition is officially underway! The topic this time around is interactive educational software for children. Complete competition rules and entry information are given in this issue. Get those computers fired up and ready to go! The prizes are fabulous! TS1000/TS1500 division 1st Runner Up - a Byte Back MD-2B Modem. TS2068/Spectrum division 1st Runner Up - a 32K Non-Volatile Memory Board. Grand Prize, Best Program Overall - a Writeman NLQ Dot Matrix Epson Compatible Printer! And that's not all! Don't forget the Fame and the Glory. All the winners will have their programs published in SYNCWARE NEWS. The best of the rest could very well find their way into one of our journals published for the TS1000 and TS2068. Get with it guys and gals. Let's hear from you soon.

New Improved

Ad agencies claim that tacking the words 'new and improved' onto your product is good for at least a moderate sales increase. Whether this is just Madison Ave. hype or not, I'm not sure. I am sure that those words, new and improved, could honestly be applied to our computers on an almost monthly bases. Even before I became Editor of SYNCWARE NEWS, I was always amazed at how many new ideas for improving software and technical applications continued to crop up for our computers. As Editor of this publication, I see something new for each Timex computer on an almost weekly basis. Even though the corporate mogols of computerdom have turned their backs on the Timex Computer line in the US, the "commoners" have proven over and over that we own more than just high-tech door stops. To prove my point, I'd like to turn your attention to two articles appearing in this issue.

CONTINUED 'IMPROVED' PAGE 24

FOR YOUR SUPPORT

Chia-Chi Chao, 73 Sullivan Drive, Moraga, CA 94556, announces that he is now selling three programs from JRC Software. Diamond Mike II, \$17.00; Great Game & Graphics Show, \$17.00; Compass, \$19.00. All prices include shipping. The games are available on tape or 5.25" Aerco disk. A complete catalog is available with SASE.

NovelSoft, 106 Seventh Street, Toronto, Ontario, Canada M8V 3B4, (416) 259-8682, announces two new pieces of software for either the TS2068 or Spectrum. ARTWORX allows you to not only draw, but paint and do graphic design on your computer. Features include pull-down menus, multiple brushes and text fonts, cut and paste, and much more. Includes sample artwork and manual. TIMACHINE is a BASIC compiler that will convert your programs into machine code to increase running speed to 200 times normal. The program will handle all BASIC and floating point except I/O. Includes demo programs and manual. Both programs include Spectrum and 2068 versions. The cost is \$19.95 each U.S. plus \$3 S&H.

Willcocks Research Consultants, 6321 W. 78th Place, Los Angeles, CA 90045, (213) 215-0780, announces the QL version of KARTIK, a family crossword game, which includes an electronic dictionary. KARTIK and TANGLE-4 are also available for TS1000/1500 computers. Write for details and prices.

Grey & Clifford Computer Products, P. O. Box 2186, Inglewood, CA 90305, (213) 759-7406, is offering software and hardware for communications on the Spectrum emulated TS2068. SPECTERM-64 is terminal emulation software which allows 64 column display without additional hardware. Price for SPECTERM-64 is \$30. When used with the new Z-SI/O hardware card, it allows use with any RS-232C modem or peripheral including Hayes compatibles. Price for the Z-SI/O is \$79.95. Write for more details.

Knighted Computers, 707 Highland Street, Fulton, New York 13069, is selling QL computers, software, and related products. Their current flyer lists the book "T/S2068 Basics and Beyond" by Sharon Z. Aker.

Fred Nachbaur, Silicon Mountain Computers, C-12, Mtn. Stn. Group Box, Nelson, BC V1L 5P1, Canada, has been hard at work under the mountain and has finally completed DUNGEON OF YMIR for the TS1500. This full featured HI-RES adventure game competes favorably with similar games for much larger machines. Presently available are V1 (requires Hunter board, other 8-16K NVM or 64K) and V2 (requires 16K RAM pack; may require slight hardware mod, supplied). V3 (for ZX81/TS1000 with CMOS 8-16K RAM) to follow soon; stay tuned. Price \$24.95 U.S.

Robert C. Fischer, producer of PRO/FILE EXTENSIONS, T/S GRADER, WORD PUZZLER, and WORD GAMES, has changed his address and can now be found at Rt. 2 Arizona St., Emerson, GA 30137

Larry Kenny, Larken Electronics, has announced that he is working on a cart-bases DOS for the Larken 2068 disk drive. Features include Spectrum compatibility, built-in NMI "snapshot" save, usage of standard tokens, and disk compatibility with other systems. Also in the works are EPROMs for use with other popular disk systems. Contact LARKEN ELECTRONICS, RR#2, Navan, Ontario K4B 1H9, Canada, to be placed on the "inform-when-ready" list, or to suggest features that YOU would like to see.

Bill Bell, 596 Cherrington Road, Westerville, OH 43081, (614) 882-3883, is offering BBDOS for the TS1000 equipped with the Aerco FD-ZX Disk Interface. BBDOS is a fully automatic, BASIC transparent, disk operating system that locates itself in the 8-16K region of RAM. It is advertised as being fully compatible with all printer interfaces. \$29.95.

Thomas B. Woods, PO Box 64, Jefferson, NH 03583, announces the availability of PRO/FILE AROS for the TS2068. The price is \$59.95. Included is the AROS cartridge, PRO/FILE instruction manual, and a special supplementary book. This updated and improved version of PRO/FILE has built-in software for any of seven different printer I/Fs. Software to allow the use of an IBM compatible keyboard via a Experimenter's Universal I/O Port (also available from Woods. Reg. \$69.95, now on sale for \$39.95) is also included. SASE for more info.

Jim Houston Interpises, 414 West Elsmere Pl., San Antonio, TX 78212, offers a shifter board for Timex computers. This allows you to use a full size keyboard and assign the extra keys with functions that normally require multiple key presses. The Shifter Board is sold bare for \$17.50 or fully assembled and tested for \$45.00. Also available are an Edge Connector board, \$55.00; a Direct Video with Inverter board, \$15.00; and a line of programs on AERCO disk, for \$29.95 each. For more information, please send a SASE to the above address.

WIDJUP Co., 1120 Merrifield S.E., Grand Rapids, MI 49507, has introduced TOURIST C for the TS2068. The program is described as an extended bank switching disassembler and SPY program which resides in BASIC. Also included is an universal printer interface driver. This program is to help in the disassembly of code irregardless of which bank the code may reside. SASE for more info.

David A. Hill, 1159 South Shore Drive #12, Holland, MI 49423, is currently publishing a newsletter dedicated to the users of the AERCO FD-68. The publication, appropriately titled "FD-68 USER", contains information on getting the most from your disk system. Topics include program conversions and modifications for disk use. Send a SASE for info.

* SMART TEXT TS-2068 *

-gives you the prettiest letters that you'll ever see from a computer. It's DATA BASE lets you edit and move data at will. It's MAIL MERGE lets you create or LOAD mail list files for PERSONAL FORM LETTERS. HEADERS, FOOTERS, BLOCK INDENT, Repeat Print reports, do invoices, outlines, AUTO LETTERHEADS, AUTO SIGN OFF! A complete ADMINISTRATIVE PACKAGE for home or office.

\$39.95 Check or M/O. Post Paid.

Specify type of Interface, and for Cassette, Micro Dr, or AERCO FD Disk. Supports all printers. 70 page illustrated manual, plus phone info service. Guaranteed.

Bill Jones, Gulf Micro Electronics, 1317 Stratford Ave, Panama City, FL 32404. 904-871-4513 Inquiries welcome.

FORUM

SYNCWARE NEWS welcomes correspondence from its readers. Some editing of letters may be necessary for brevity or to enhance clarity.

Monitor Compatibility

Dear Editor,

There are now a great many excellent buys available on monochrome monitors. I recently purchased one from Jameco Electronics for \$38.00 including case, power supply, and keyboard. Unfortunately, I find that many of these, including mine, require that the sync signals be separated from the video signals rather than the composite video signal as available from the Timex.

A circuit was shown in Volume #1 of SyncWare News that would separate the sync pulses from the video, but I'm sure many of your readers no longer have that issue available. Also, as in my own case, many of these monitors require that the horizontal sync pulses be separated from the vertical sync pulses. I'm sure many readers would appreciate instructions on how to wire such a set-up.

David Thomas
Millbrook, AL

You are correct in your observation that there are many good 'bargains' on monitors at the moment. However, as you have discovered, most of these will not work with a standard Timex computer. These surplus monitors usually require TTL video and sync signals. Unless the computer has been designed to provide these signals, they are almost impossible to derive otherwise. The Timex computers have been designed to provide the user with composite signals only. If you own a TS2068 (your letter never stated which computer you had), companies such as AERCO have boards which produce TTL level signals which are

compatible with some of these surplus monitors. This is possible because the computer already supplies most of what is needed at the rear expansion buss.

The circuit you refer to in Volume #1 of SyncWare News (pg. 62 in the reprinted version) on casual examination does appear to provide separate sync signals. In reality, all it does is to prevent sync signal inversion when the video is reversed. A circuit to actually separate the sync pulses would be much more involved. Timing circuits, and the like, would be necessary to do the job properly. In talking to our resident technical guru, Fred Nachbaur, it was decided that a properly designed and functional circuit to separate video, horizontal sync, and vertical sync pulses would be no small task. If anyone out there knows of such a circuit we would be more than happy to pass the word along to the rest of our readers.

In the meantime, I can only suggest to our readers to read ALL the small print before laying down your hard earned cash for a monitor 'bargin'. Be certain it is a composite type unless you KNOW you can use other style monitors.--jm

Pascal Help Wanted

Dear Editor,

I am a fairly recent member of the "SyncWare News" family of readers, and I turn to our group of readers out there for help in solving my current problem.

I have recently become interested in running "PASCAL" programs on my TS2068. I would appreciate any information on a PASCAL Compiler for use with the TS2068 that any of our advertisers or readers might have to offer.

I also have a Zebra Systems Delux Spectrum Emulator, so any thing that will work with that would be of interest to me also.

Can anybody out there help me?

Melvin B. Hoyt
4409-49th St. Moline, IL 61265

The information I can locate in regard to PASCAL software is very minimal. Two companies, Hawk Wild Software and English Micro Connection, at one time listed PASCAL programs in their sales catalogs. However, neither of these companies is presently in business. Knighted Computers, (315) 593-8219, did buy out the remaining stock of English Micro Connection when they closed their doors due to health reasons. Unfortunately, the stock that remained contained no PASCAL software. I can only suggest that you pick up one of the English computer magazines such as ZX Computing Monthly. Contact a few of the distributors listed and see what they have to offer. As for the Spectrum software, I'd suggest that you take a few moments of your time and write to a few of our many advertisers or folks that appear in the 'For Your Support' column. What they have to offer just might surprise you. Please include a SASE as a courtesy as many of these suppliers operate on a limited budget.

If any of our readers know of or are selling programs that might be of interest to Mr. Hoyt, please feel free to drop him a line. We have included his address for your convenience.--jm

☆ **SCLD** ☆

Revive Your DEAD TS2068

OR

**Make Your TS2068
INDESTRUCTIBLE!!**

100 Chips- First come basis

\$20 ea. \$18 ea. for 2 or more

**W/instructions
optional 68PCC socket available**

**cks to: CATS USER GROUP
Ruth Fegley
5736 Ridgedale Rd./side
Baltimore, MD 21209
(301) 542-2549**

Chips Are HERE!!

2068 Art Revisited

Dear Editor,

On page 17 of the Mar/Apr 1985 issue of SYNCWARE NEWS you published a program by David Kulp entitled 2068 ART.

I found that if you add:

65 INK RND * 6

And change line 200 so that it reads:

200 BORDER 0: PAPER 0

you get quite a different display.

Charles Fursdon, Florida

Wanted: 2068 Graphics

Dear Editor,

We would like to request that your magazine feature more 2068 Business Graphics, Logo, and Letterhead Design software. Also of interest to us would be programs related to X,Y,Z Axis data base plots, Ham Radio, and SWL.

We look forward to your publication continuing to support 2068 applications.

Paul Dolengewicz
EDP Research & Development Co.
Westbrook, CT

Thank you for your letter. This month's issue of SYNCWARE NEWS has a nifty label making program that you might find some use for, either as is or modified. We have every intention to continue supporting ALL the Timex computers until there are no more articles to print or we no longer have any subscribers. Rest assured, if we get any articles on the above topics, we'll print them.--jm

Keyboard Troubles

Dear Editor,

Yesterday my Spectrum Plus dropped dead. That is, all input from the keyboard was lost. Surgery to the computer seemed to be the only recourse. I took apart the computer and cleaned the keyboard 'tails' with a gold cleaner and preservative. Please warn all your readers to NEVER do this. The cleaner removed not only the dirt but the copper traces on the keyboard as well!

I have since gotten the computer to work again by shortening the tail and re-connecting the keyboard to the main chassis. The 'modified' tail is so short though, I'm still having problems. Can anyone out there help me by coming up with a new innterconnect between the keyboard and the computer?

Stan Nagrod
Albertson Communications Mamaroneck, NY

That little ribbon cable between the computer and the keyboard has historically been a real pain. Because the traces on it are so fragile, any flexing of the cable can easily break them. This is especially true if you frequently dis-assemble your computer. I have never worked with a Spectrum Plus so can only make general suggestions.

First, you might try physically removing the

connector on the computer board. Cut back the keyboard tail until you find an area that is solid both electrically and mechanically. Insert the cable into the connector you previously removed from the computer board. Using standard ribbon cable available from Radio Shack, etc., run wires from the connector to the holes where it was removed. Watch for solder bridges, and tape all exposed solder joints at the connector before putting the computer back together.

Second, several suppliers offer improved add-on keyboards for the TS2068. Perhaps one of these could replace your old keyboard altogether. Along those same lines, Tom Woods sells an IN/OUT Port for the 2068 that I've seen hooked up and running with an IBM style keyboard. These surplus keyboards are currently available for IBM clones or as replacements. Perhaps something in this vain would solve your problem and improve your computer to boot.--jm

Ink Jet Woes

Dear Editor,

I have mated an Olivetti PR2300 Ink Jet Printer to my TS1000 via a Memotech Centronics I/F. The printer will LLIST and LPRINT fine. It will also COPY okay as long as there are no graphic symbols on the screen. However, in spite of everything I've tried, I can not get the printer to function in the Plotter Mode.

Have you any ideas that might get the Plotter Mode to work?

Robert Haley
Wooster, Ohio

From the rest of your letter, it would in fact seem that you have tried everything to get the Plotter Mode to work. Being unfamiliar with the Olivetti printer I can offer no ideas as far as printer codes go. We will print your letter in hope that another reader might have some ideas to offer.

The WIDJUP Co., 1120 Merrifield S.E., Grand Rapids, MI 49507, is currently offering a printer utility for the TS2068 and the Olivetti PR2300. We have not had an opportunity to review their software. It's possible that by contacting them they might shed some light on your problem. Although they haven't advertised it, they may have a TS1000 version of the program. Write and see, then let us know.--jm

More on Mandelplot

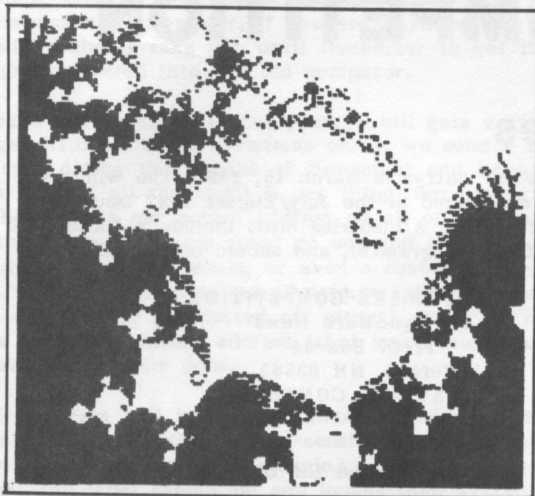
Dear Editor,

I really enjoyed the Mandelplot article by Fred Nachbaur. I have a 2068 with a Spectrum switch and I found that the program would run as is in the Spectrum mode.

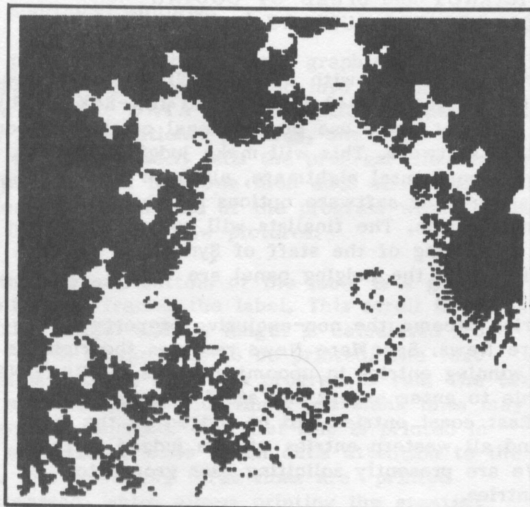
I have been interested in fractals for about two years. I don't fully understand them, but they are fascinating. I printed the enclosed mandel plots using the preview mode of the program, saving the screen\$ and transferring it to Pix Fix. (I have trouble dumping screens to my Olivetti printer.) After spending 2.5 days developing the Gemini-10 picture, I saved the data array and started trying a few modifications. Below is the listing for the first picture (fig. F-1). I then used the data from the Gemini-10 character array to feed another fractal.

The last number in line 4090 is only to adjust the up/down position of the picture on the screen.

```
4030 LET M$=CHR$ MX
4060 FOR M=1 TO 188
4070 LET Y=M
4080 FOR N=1 TO 189
4090 IF P$(M,N)>M$ THEN PLOT 1000*((ASN (N/1000)-
(INT ASN (N/1000)))),(1000*((ACS (Y/1000)-(INT ACS
(Y/1000)))))-300
```



Produced using the character array from the centronics (Gemini) Mandel-Plot and a modified line number 4090.



Produced using the original line number 4090.

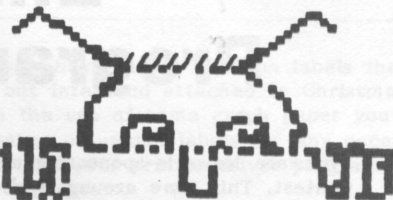
A picture previewed using these changes will take a couple of hours to develop.

George Lipscomb
TSgt, USAF

SWN Wants ART

Periodically as we paste up SYNCWARE NEWS, we find ourselves with a half a column or so with nothing small enough to fit. We are looking for some cartoonists to send us computer and/or electronic related original cartoons to fill these occasional

voids. We'll pay you \$5.00 when we use one in the magazine. To see if people read everything in this column, or just the first few lines, we'll also accept short programs too. Things along the lines of TYD*BYTS, only shorter. The pay rate will be the same for these as the cartoons.



BUG ALERT

Dave Youngquist writes to alert us that a greeble has found its way into the STATIC RAM CARTRIDGE article that appeared in the May/June 1986 (Volume 3:5) issue of SyncWare News. On page 10, assembly step #9, the text erroneously reads "...connect to pin 27 of Ea and Eb, at 2e and 2f." The corrected instructions should be "...connect to pin 27 of Ea and Eb, at 2j." Mr Youngquist states that the error would not have caused any damage to the computer or the RAM board. The wiring in the article originally had pin 27 of both RAMs connected to ROSCS NOT. The correction connects these pins to WR NOT as they should be.

WOW! AT LAST!! A VERY AFFORDABLE COMPUTER AT A VERY AFFORDABLE PRICE

POWERFUL FULLY PROGRAMMABLE WITH 2K OF MEMORY—PORTABLE—5 7/8 x 1-3/8 INCH MODULE SINGLE-KEY ENTRY COMMANDS—DURABLE 40 KEY MEMBRANE TYPE KEYBOARD—280A BASED FOUR CHIP DESIGN—EDUCATIONAL—UNIQUE SYNTAX-CHECK REPORT CODES FOR ERROR IDENTITY—GRAPH DRAWING AND ANIMATED DISPLAY—ACCURATE TO 5 1/2 DECIMAL PLACES FOR FULL RANGE MATH AND SCIENTIFIC FUNCTIONS—AT AN AFFORDABLE PRICE.

WE CANNOT TELL YOU THE MAKE OF THE COMPUTER BUT IT WAS MADE BY A FAMOUS WATCH COMPANY. THEY USED TO SELL FOR \$99.95.

WE BOUGHT OUT WHAT THE FACTORY HAD LEFT IN STOCK AND HAD TO REMOVE THE LABELS. THESE UNITS ARE UNPACKAGED. LESS THE \$V WALL ADAPTER AND MANUAL. BECAUSE THIS IS A DISCONTINUED ITEM THERE IS NO WARRANTY.

GET THEM WHILE THEY LAST LIMITED SUPPLY

BUY 1st UNIT FOR \$19.95 BUY 2nd FOR \$16.95 9V DC WALL ADAPTOR \$4.95
BUY THE 3rd UNIT (non operating for parts) \$10.95 MANUAL (over 150 pages) \$2.95

See September 1984 issue of 73 for TIMEX/RITTY article

CHIP BONANZA (AT THESE PRICES THEY ARE A STEAL)	
2708	\$1.00 EA OR 10 FOR \$ 9.00
2710	\$2.25 EA OR 10 FOR \$20.00
2732	\$3.25 EA OR 10 FOR \$30.00
2784	\$4.00 EA OR 10 FOR \$36.00
27138	\$7.00 EA OR 10 FOR \$60.00
6802	\$4.95 EA OR 10 FOR \$49.50
6810	\$1.95 EA OR 10 FOR \$19.50
68A08	(REG. \$19.95) \$5.95 EA OR 10 FOR \$59.50
68A21	(REG. \$9.95) \$2.95 EA OR 10 FOR \$29.50
4116	(or equivalent) 8 FOR \$ 5.99
4134	9 FOR \$12.95
XR-2211	(SPECIAL) 8 2.95
TMS 9800NL MICRO-P 64 PIN 8 BIT D/B + 16 BIT CPU	4 4.95
TMS 9801NL MICRO-P PSI	2 2.95
TIM 9801NL MICRO-P CLOCK GEN. AND DRIVER	2 5.95
TMS 9818NL MICRO-P COLOR GRAPHICS AND DIGITAL	2 9.95
KEYBOARD (884) 48 KEYS MEASURE 4 x 10 (H)XENK	5 9.95
APPLE II and APPLE II+ COMPUTER MAINFRAMES (fully populated) \$150.	
Power supply, case and keyboard, separately available. Call or Write	
Unit as described above, fully assembled & tested ... \$295.00 plus shipping	
APPLE POWER SUPPLIES ... \$29.95	
Cassette Software:	
I have cassette software ... send for list (reg. \$9.95-\$14.95)	
Hal's Special Prices: 1 for \$3.95; 2 for \$10.95; 5 for \$19.50; 10 for \$39.50; 20 for \$50.00	
... or let Hal select 25 different cassettes (my choice) at 25 for \$50.00!!!	

SPECIALS

IBM Compatible Computers

Flip-Top Case, Keyboard, Power Supply, and Motherboard. Up and Running ... \$399.00 complete (plus \$5.00 for shipping)

FCC Approved

CALL FOR DETAILS

LNA

90° Kelvin, self-contained electronic polarizer, and feedhorn ... \$59.95 each or 2 for \$100.00 (plus \$5.00 for shipping)

SHIPPING INFORMATION: PLEASE INCLUDE 10% OF ORDER FOR SHIPPING AND HANDLING CHARGES (MINIMUM \$2.50, MAXIMUM \$10). CANADIAN ORDERS, ADD \$7.50 IN US FUNDS. MICHIGAN RESIDENTS ADD 4% SALES TAX. FOR FREE FLYER, SEND 22¢ STAMP OR BARE.

HAL-TRONIX, INC.
P.O. BOX 1101 DEPT. N
12671 DIX-TOLEDO HWY
SOUTHGATE, MICH. 48195 PHONE (313) 281-7773



"HAL"
HAROLD C. NOWLAND
W8ZXH

announcing The SyncWare News 2nd (sort of) Annual Programmers' COMPETITION

Syncware News is sponsoring another Programmer's Contest. This time around, we're looking for the best in interactive educational software for children. The winners and the best of the rest will be considered for publication.

Interactive educational software, in this case, is not limited to Addition or Spelling games for first graders, although these are surely welcome. We'll also consider educational simulations, demos, and the like. Artificial Intelligence can come into play here, too. A simulation program that would let a student see how a saw mill works, for instance, is an educational program. It becomes interactive when it asks questions about what the child has seen demonstrated and rewards for the appropriate response.

Prizes!

GRAND PRIZE - Best Program Over All
Writeman NLQ Dot Matrix Printer - features include 160 cps; 8K buffer; Epson compatible print commands; 9 wire print head; Standard, Italic, and Proportional print characters; down-loadable graphic characters; Retail Value, \$400.00

FIRST RUNNER UP - TS2068 DIVISION
32K Non-Volatile RAM Cartridge Board - features include 32K of non-volatile RAM which can be mapped to reside in either the Cartridge Bank or the EXROM Bank; Battery Backup to prevent data loss; runs from 100% battery power or will draw power from the 2068 while the 2068 is powered up; Write protect switch; Easily modified Chunk specifications; 32K NVM can be used to mask EXROM or contain LROS and AROS programs; Retail Value, \$110.00

FIRST RUNNER UP - TS1000/TS1500 DIVISION
Byte-Back MD-2B Modem - features include the ability to upload and download text to and from tape and memory; 300 baud - word length, parity, etc. selectable; direct connects to telephone line; RS-232 port; Retail Value, \$129.00

Rules

Deadline for entrys is March 15, 1987. The winners will be announced in the July/August 1987 issue of SyncWare News. All entries must include a CASSETTE TAPE of the program(s), and should be mailed to:

SOFTWARE COMPETITION
c/o SyncWare News
P. O. Box 64
Jefferson, NH 03583
ATTEN: CONTEST

Entries will be judged on the basis of:

- 1) MEETS CONTEST CRITERIA - is it educational?
- 2) DOCUMENTATION
- 3) EASILY MODIFIED - can I easily insert my child's spelling words into a spelling program?
- 4) AESTHETICS
- 5) EFFICIENCY and SPEED OF CODING
- 6) TS2040 PRINTER SUPPORT (but not mandatory)

All entries must work with NO ADDITIONAL HARDWARE (except the TS2040 printer, and 16K-64K RAM). Entries may not make use of additional commercial or proprietary software. This will make judging entries less of a monumental nightmare, although support of other hardware or software options will be taken into consideration. The finalists will be judged by a panel consisting of the staff of SyncWare News. All decisions of the judging panel are final.

All entries become the non-exclusive property of SyncWare News. SyncWare News reserves the right to publish winning entries in upcoming issues. EVERYONE is eligible to enter except the staff of SyncWare News. East coast entries will be judged on the West coast and all western entries will be judged in the East. We are presently soliciting user groups to judge entries.

Remember, you can't win if you don't enter. Get started right away!

Super Prizes, Fame & Glory can be YOURS!

2068 LABEL MAKER

Created by Greg Dertz for Tandy Color Computer II and Modified by Bob Swoger for the SINCLAIR 2068 TASMAN B CPI, and the LEGEND 1080 Printer. A gift from the both of us to the MOTOROLA MICROCOMPUTER CLUB and the readers of SYNCWARE NEWS.

Merry Christmas to all! It may seem like I am jumping the gun a little by wishing you a Merry Christmas now. However, if you are anything like me it will probably take you until December to get this program entered into the old computer.

Around Christmas time, our postage bill gets very large. With all those Christmas cards, we send a lot of mail during the months of November and December. The cost of all that mail is one thing, but having to hand write our return address on all of that mail is too much. Sure, we could go out and get some ready-made return labels, or even a rubber stamp, but those do not show the Christmas spirit (or cause the old 2068 to get dusted off either). What we need is a Christmas return address label maker and that's what this program does.

This program will make any number of return address labels on 15/16 inch pressure sensitive labels. The labels must be "one-up," meaning only one across the page. The label cannot be any longer than 3 1/2 inches. The text of the label is variable and entered when the program is run. In addition to the text of the label, the program also adds a small, graphically printed picture to the label. Currently, the program has four such pictures in its library, but the program may be easily modified to add additional pictures.

The picture is printed using a graphic area of 32 X 40 dots. The picture is placed into the program using READ - DATA statement pairs in lines 250 to 260. To add additional pictures, one must change line 40, MP=4, which tells the program the number of pictures to look for. One then adds additional DATA statements to the end of the program which contains the DATA for the new pictures.

At the top and bottom of the label is a graphic scroll which frames the label. This scroll is of a variable length and its length is calculated from the length of the text to be printed. The text itself is entered when the program is run. The text has a maximum of four lines, but blank lines may be entered if required. The first line is printed in the emphasized mode which calls attention to the line. The remaining three lines are printed compressed, which allows printing the greatest amount of information on the line. The program was written for the LEGEND 1080 printer but should run on any EPSON RX-80 with GRAPH TRAX compatible printer with only a slight modification to the scroll length formula do to the fact that LEGEND 1080 is a 142 column printer while most others can only do 132 columns.

In addition to return address labels, we have also used the program to print out Christmas gift tags. We enter the four lines as follows:

A VERY MERRY CHRISTMAS
TO:
FROM:

A VERY MERRY CHRISTMAS
TO: All Our READERS
FROM: The Staff and Contributors
of SyncWare News

CHRISTMAS RETURN LABEL
And Gift Tag Maker
By Greg Dertz
and Bob Swoger

This way, we now have blank, stick-on labels that may be filled out later and attached to Christmas packages. With the use of some graph paper you can create any number of unique labels for any occasion. Then, just load your printer driver program followed by this program.

Happy label making.

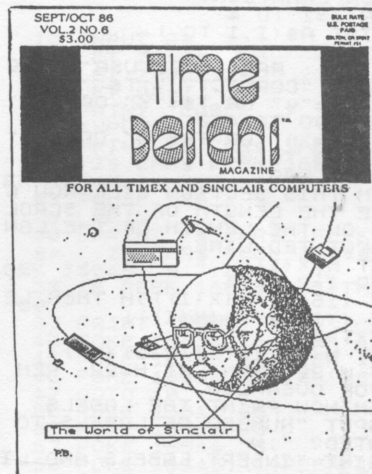
EDITOR'S NOTE:

For those of you that have a modem and are using SMART II Software, Messrs. Dertz and Swoger invite you to call the Motorola Microcomputer Club's computer and download this program to your TS2068. Set your modem for 7/E/1S and have SMART II's buffer closed and empty, running in hex. Call the computer at (312) 576-7072. Hit the <ENTER> key a couple times until you get ->. At the prompt LOG:, hit the <ENTER> key again. The next prompt is ACCOUNT OR NAME:. Enter the word "account". You will then be asked for a PASSWORD:; enter the word "guest". The program will automatically open the buffer, download, and then close the buffer. When GOODBYE appears on your screen, hang up. Remember that you are a guest in this system, conduct yourselves as such. 'Nuff said.-jm

America's BEST Timex/Sinclair Magazine
is now the BIGGEST!

TIME DESIGNS MAGAZINE

A valuable resource for your TS 2068, TS 1000 (zx81),
TS 1500, Sinclair Spectrum and Sinclair QL.



Largest Circulation
of any TS specific
Magazine.
Largest Page Count.
More Program Listings.
Largest TS Classified.
Many More Features!

Now
In Our
THIRD
Year!

Still
Only
\$15 Year
Six Issues
Delivered on Time!

DON'T DELAY...TRY A SUBSCRIPTION TODAY!

Send Name, Address, along with check/MO OR Credit Card No./Exp. date to:

TIME DESIGNS MAGAZINE COMPANY

29722 Hull Rd., Colton, OR 97017 (503) 824-2658

```

10 REM CHRISTMAS LABEL GENERAT
OR
20 REM WRITTEN BY GREGORY A.
DELTZ & ROBERT E. SWOGER ON
OCTOBER 23, 1985
40 LET MP=4
50 DIM A$(4,36): DIM S(12): DI
M K(MP,4,40): DIM X(4)
60 REM THIS IS THE DATA FOR TH
E UPPER AND LOWER SCROLL
70 FOR I=1 TO 12
80 READ S(I)
90 NEXT I
100 DATA 64,32,48,24,28,28,28,2
8,24,48,32,64
110 LET LC=1
120 CLS: PRINT " WELCOME TO
THE CHRISTMAS"
130 PRINT " RETURN ADDRESS LABE
L PROGRAM."
140 PRINT " I WILL FIRST ASK FOR
YOUR NAME AND ADDRESS(4 LINES
MAX.), THEN "
150 PRINT " PRINT OUT THE NUMBE
R OF RETURN "
170 PRINT " ADDRESS LABELS REQUE
STED."
180 PRINT " THE FIRST LINE OF TH
E LABEL WILL "
190 PRINT " BE PRINTED EMPHASIZE
D, THE "
200 PRINT " REMAINING THREE LINE
S WILL BE "
210 PRINT " PRINTED COMPRESSED."
220 PRINT " TO USE ME YOU MUST
HAVE LABELS "
230 PRINT " THAT ARE 1 TO THE IN
CH AND ARE "
240 PRINT " ONE UP ON THE SHEET.
"
245 REM WHILE THE USER IS READ
ING THE ABOVE STUFF, START READ
ING THE DATA FOR THE PICTURES.
250 FOR I=1 TO MP
260 FOR J=1 TO 4
270 FOR X=1 TO 40
280 READ K(I,J,X)
290 NEXT X: NEXT J: NEXT I
300 GO SUB 900
310 CLS: PRINT "EXAMPLE:",
MERRY CHRISTMAS", "John, Mary, J
im, & Spot", "1234 Any Street": P
RINT "This Town, IL 60000"
320 REM NOW ASK FOR THE RETURN
ADDRESS INFO
330 FOR I=1 TO 4
340 PRINT " LINE ";I;"? ";: INP
UT G$
350 IF G$="" THEN GO TO 365
360 LET X(I)=LEN G$: LET A$(I)=
G$
365 PRINT A$(I,1 TO )
370 NEXT I
375 PAUSE 150
380 CLS: REM NOW ECHO THE INPU
T FOR CONFIRMATION.
390 FOR I=1 TO 4
400 PRINT A$(I,1 TO )
410 NEXT I
420 PRINT " RANDOMIZE USR 64719
430 INPUT "CORRECT? ";T$
440 IF T$="Y" OR T$="Y" OR T$="
YES" THEN GO TO 470
450 IF T$="N" OR T$="N" OR T$="
NO" THEN GO TO 330
460 GO TO 390
465 REM THIS SECTION WILL NOW C
ALCULATE THE LENGTH OF THE SCROL
L BASED ON THE LENGTH OF THE LONG
EST INPUTTED LINE.
470 LET H=X(1)
480 FOR I=2 TO 4
490 IF (15/27)*(X(I))>H THEN LE
T H=INT ((15/27)*(X(I)))
500 NEXT I
510 LET H=5+(INT ((2/3)*H))
520 IF H>22 THEN LET H=22: REM
CHECK FOR OVERFLOW
525 REM NOW PRINT THE LABELS
530 INPUT "NUMBER OF LABELS TO
BE PRINTED? ";N
540 PRINT "INSERT LABELS AND LI
NE UP THE "
550 PRINT "FIRST LABEL.",

```

```

580 REM NOW CLEAR THE PRINT ER
AND SET LINE SPACING TO 8/72 INC
H SO THAT THERE WILL NOT BE SPAC
ES BETWEEN THE LINE S OF THE PIC
TURE
581 LPRINT CHR$(27);CHR$(27);"@";
CHR$(27);CHR$(27);"A";CHR$(27);CHR$(
8);
585 REM NOW THEN PRINT ALL OF T
HE LABELS
590 FOR Z=1 TO N
600 PRINT "LABEL NO. ";Z;
610 GO SUB 840: REM PRINT THE U
PPER SCROLL
620 FOR L=1 TO 4
630 LPRINT CHR$(27);CHR$(27);"L";
CHR$(80);CHR$(27);CHR$(0);
640 FOR C=1 TO 40: REM PRINT PI
CTURE LINE
650 LPRINT CHR$(27);CHR$(K(LC,L
,C));CHR$(27);CHR$(K(LC,L,C));
670 NEXT C
680 IF L=1 THEN LPRINT CHR$(27);
CHR$(27);"E";: REM PRINT THE FIRS
T LINE EMPHASIZED
690 IF L=2 THEN LPRINT CHR$(27);
CHR$(27);"F";CHR$(27);CHR$(15);CHR$(
27);CHR$(27);"G";: REM PRINT AL
L OTHER LINE S COMPRESSED
700 LPRINT A$(L)
710 NEXT L
720 GO SUB 840: REM NOW PRINT T
HE LOWER SCROLL
725 LET LC=LC+1: REM INCREMENT
THE PICTURE
726 IF LC=MP+1 THEN LET LC=1
730 LPRINT: LPRINT: LPRINT
740 LPRINT CHR$(27);CHR$(27);"H";
: RESTORE: PRINT "PRINTED": NEX
T Z
790 LPRINT CHR$(27);CHR$(27);"@";
: REM NOW DONE, SO RESET THE PRI
NTER
800 INPUT "WOULD YOU LIKE MORE
LABELS? ";T$
810 IF T$="Y" OR T$="Y" OR T$="
YES" THEN GO TO 380
820 IF T$="N" OR T$="N" OR T$="
NO" THEN PRINT "O.K. THANK YOU
FOR USING ME.": STOP
830 GO TO 800
840 LPRINT CHR$(27);CHR$(27);"H";
CHR$(27);CHR$(27);"K";CHR$(27);CHR$(
12*H);CHR$(27);CHR$(0);
850 FOR B=1 TO H
860 FOR I=1 TO 12
870 LPRINT CHR$(27);CHR$(S(I));
880 NEXT I: NEXT B: LPRINT: RE
TURN
900 PRINT AT 21,4;"HIT <ENTER>
TO CONTINUE "
910 LET T$=INKEY$: IF T$="" THE
N GO TO 910
920 RETURN
1000 REM TREE ARRAY LC=1
1010 DATA 0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,1,7,63,255
1020 DATA 255,63,7,1,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,
1030 DATA 0,0,0,0,0,0,0,0,0,0,0,0,
0,1,3,7,79,223,255,255,255
1040 DATA 255,255,255,223,79,7,3
,1,0,0,0,0,0,0,0,0,0,0,0,
1050 DATA 0,0,0,0,0,0,0,0,0,1,35
,239,255,255,255,255,255,255,255
255
1060 DATA 255,255,255,255,255,25
5,255,255,239,35,1,0,0,0,0,0,0,0
,0
1070 DATA 0,0,0,0,0,16,48,112,24
0,240,240,240,240,240,240,240,24
0,255,255,255
1080 DATA 255,255,255,240,240,24
0,240,240,240,240,240,112,48
,16,0,0,0,0,0
1090 REM BELLS ARRAY LC=2
1100 DATA 0,0,0,0,0,0,0,0,0,0,0,
0,1,3,3,7,3,0,7,7
1110 DATA 15,15,31,31,15,15,7,7,
0,0,0,0,0,0,0,0,0,0,0,0
1120 DATA 0,2,3,3,3,3,7,15,31,63
,127,255,255,255,255,248,7,255,2
55,255

```




```

1130 DATA 255,255,255,255,255,255,25
5,255,255,255,7,0,0,0,0,0,0,0,0,
0,0
1140 DATA 0,0,0,152,252,252,248.
240,248,252,254,255,255,254,241
15,255,255,255,255
1150 DATA 255,255,255,255,255,25
5,255,255,255,255,15,1,0,0,0,0,
0,0,0
1160 DATA 0,0,0,0,0,0,0,0,0,0,0,
64,64,192,192,192,192,192,192,19
2
1170 DATA 192,192,192,192,192,19
2,192,216,252,252,216,192,192,64
,64,0,0,0,0
1180 REM STOCKING ARRAY LC=3
1190 DATA 0,7,2,1,7,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0
1200 DATA 0,0,0,0,0,0,0,0,0,1,1,29
,51,62,12,0,0,0,0,0
1210 DATA 0,128,0,0,128,0,60,36,
36,60,0,1,1,1,1,0,0,0,0
1220 DATA 0,0,0,48,44,67,135,135
,15,31,31,159,127,15,3,0,0,0,0
1230 DATA 0,0,0,0,0,0,0,0,0,0,0,
240,80,16,16,0,15,1,1,1
1240 DATA 0,0,0,0,0,0,129,227,255,
255,255,255,255,255,255,255,
63,12,0,0

```



CHRISTMAS RETURN LABEL
And Gift Tag Maker
By Greg Dertz
and Bob Swoger

A VERY MERRY CHRISTMAS
TO: All Our READERS
FROM: The Staff and Contributors
of SyncWare News

CHRISTMAS RETURN LABEL
And Gift Tag Maker
By Greg Dertz
and Bob Swager

TS1000 ADVANCED CALCULATOR PROGRAM

My work is primarily applied math and engineering design. In addition to using micro and main frame computers, much of the work is 'one time thru' evaluation of complex formulas and expressions. Prior to developing the programs presented herein, this was done with scientific calculators and hand written 'engineering notes' which were passed on to the client. Use of CALPR-1 below greatly increased the accuracy and efficiency of this work and the print-out replaced the need for hand written notes. I predict anyone presently doing scientific computations 'manually' will find this concept helpful.

These programs convert the TS-1000/1500 computer (with 16 col., 32 col. or an 80 col. printer) into a superior printing scientific calculator which saves the last 25 formulas and 900 variables for reference and/or use in future expressions. The enhanced version, CALPR-1, provides for labeling of all expressions and variables for improved calculation notes. Poking to modify memory enables this to be done without formal use of arrays.

CALPR-0 uses no screen prompts. The writer has used it with a computer, 16 col. printer, battery and no display as a convenient portable unit. The program is started with GOTO 10, which preserves past variables and expression strings (formulas). The screen, if used, shows the computer awaiting a string input, which may include previous results (variables), functions, numbers, etc. Upon inputting your next expression, it and the result (with a two character notation) appears on the screen. After a two second pause, they are also sent to the printer. The next string is then called for and the cycle repeats. Strings A\$ thru Y\$ are preserved and the results are labeled A0 thru YZ (900 permutations).

SYNTACTIC SUM = 35968

CALPR-0 BY D.L. DEAN A PRINTING
SCIENTIFIC CALCULATOR PROGRAM
WITH EXTENDED MEMORY

```

10 INPUT A$
15 LET Z$=A$
20 LET Z=VAL Z$
22 LET A0=Z
25 PRINT CHR$ (PEEK 16514);"Z$="
";Z$,CHR$ PEEK 16514;CHR$ PEEK
16545;"="";Z
27 PAUSE 120
29 LPRINT
30 LPRINT CHR$ (PEEK 16514);"Z$="
";Z$,CHR$ PEEK 16514;CHR$ PEEK
16545;"="";Z
35 IF PEEK 16514>=62 THEN GOTO
50
40 POKE 16514,(1+PEEK 16514)
45 POKE 16525,(1+PEEK 16525)
46 POKE 16544,1+PEEK 16544
47 GOTO 10
50 POKE 16514,38
55 POKE 16525,38
57 POKE 16544,38
58 POKE 16545,(1+PEEK 16545)
59 IF PEEK 16545>63 THEN POKE
16545,28
60 GOTO 10
80 PRINT "CALPR-0 BY D.L. DEAN
A PRINTING SCIENTIFIC CALCULATO
R PROGRAM WITH EXTENDED MEMORY

```

SYNTACTIC SUM = 33136

Syntactic Sum

The 'SYNTACTIC SUM' shown after the programs is a convenient parity check which was required by the editors of the now defunct Sinclair oriented news-letter SYNTAX ZX80, a publication of the Harvard Group. It is apparently a code sum of all characters in the program listing. Matching this number insures that it has been 'punched in' correctly (two canceling errors which defeat the check sum is most unlikely). This sum was obtained by running a 24 byte MC program (AF 47 67 6F E5 2A 0C 40 11 7D 40 2B 4E E3 09 E3 E5 ED 52 E1 20 F5 C1 C9 in hex) which was poked in above a lowered RAM TOP before loading or typing in the program to be tested. For example, with 16K RAM, lower RAM TOP to 32743 (POKE 16389,127 & POKE 16388,231), place MC bytes from 32744 to 32767 and get check sum with command PRINT USR 32744. Use the hex loader of your choice or, for so short a program manual conversion to decimal code and 24 POKEs isn't prohibitive. Incidentally, the sum for CALPR-1 requires use of inverted characters for EXP in line 37 and the token AND in line 80. (Syntactic Sum program reprinted by permission of the Harvard Group.-ed.)

```

1 REM SYNTACTIC SUM LOADER
2 REM 16K VERSION
3 REM PART ONE
10 POKE 16389,127
15 POKE 16388,231
20 NEW

```

```

1 REM SYNTACTIC SUM LOADER
2 REM 16K VERSION
3 REM PART TWO
10 FOR I=1 TO 24
15 SCROLL
20 PRINT "INPUT NO. ";I,
25 INPUT A$
30 PRINT A$
35 POKE I+32743,(16*CODE A$+CO
DE A$(2)-476)
40 NEXT I

```

INPUT NO. 1	AF	INPUT NO. 13	4E
INPUT NO. 2	47	INPUT NO. 14	E3
INPUT NO. 3	67	INPUT NO. 15	09
INPUT NO. 4	6F	INPUT NO. 16	E3
INPUT NO. 5	E5	INPUT NO. 17	E5
INPUT NO. 6	2A	INPUT NO. 18	ED
INPUT NO. 7	0C	INPUT NO. 19	52
INPUT NO. 8	40	INPUT NO. 20	E1
INPUT NO. 9	11	INPUT NO. 21	20
INPUT NO. 10	7D	INPUT NO. 22	F5
INPUT NO. 11	40	INPUT NO. 23	C1
INPUT NO. 12	2B	INPUT NO. 24	C9

Now at last...

The FootePrint Printer Interface

- for Centronics parallel printers
- works in both 2068 and Spectrum mode
- compatible with OS-64 & Spectrum emulators
- EPROM socket and on/off switch on board
- works with both Tasman and Aerco driver software
- plugs into cartridge dock—door completely closes with cable running back under computer
- frees up rear edge connector allowing other peripherals to be used; less chance of a crash
- print driver software for LPRINT, LLIST, and COPY included for 2068 and Spectrum modes

FootePrint Interface w/software & cable . \$45⁰⁰

FootePrint with OS-64 option included . \$65⁰⁰

Bare board & instructions only . \$15⁰⁰

Cable only for use with bare board . \$15⁰⁰

All prices are pre-paid and include shipping charges.

FOOTE SOFTWARE

P. O. Box 14655 — Gainesville, FL 32604
904/462-1086 (6 pm - 9 pm EDT)

SOFTSYNC TS2068 Inventory Liquidation

Zebra Systems has just completed a special purchase of all the Timex/Sinclair product left in Softsync's warehouse in order to bring you these fine programs at special low prices. Softsync started in the Timex market with excellent products. But they now focus on IBM, Commodore and other large markets, where they sell products like the Personal Accountant for much more money.

All product is new and packaged with documentation in Softsync's attractive four-color boxes. Quantities are limited, so act now. These products list at \$19.95 to \$24.95 and are in our regular catalog for \$17.95 to \$22.95. But now our liquidation sale prices are as follows:

SALE PRICES:

1 for \$10, 2 for \$19, 3 for \$27,
4 for \$34, 5 for \$40, all 6 for \$45.

Zeus Assembler	Cat# SS05
Zeus Monitor and Dissassembler	Cat# SS06
Cyberzone	Cat# SS02
Personal Accountant	Cat# SS04
Voice Chess	Cat# SS01
Guplman	Cat# SS03

Ordering Instructions: Include \$3.00 S&H. VISA/MC Accepted. Zebra Systems, Inc., 78-06 Jamaica Ave., Woodhaven, NY 11421, (718) 296-2385.

2068 WINDOWS

by Paul Bingham

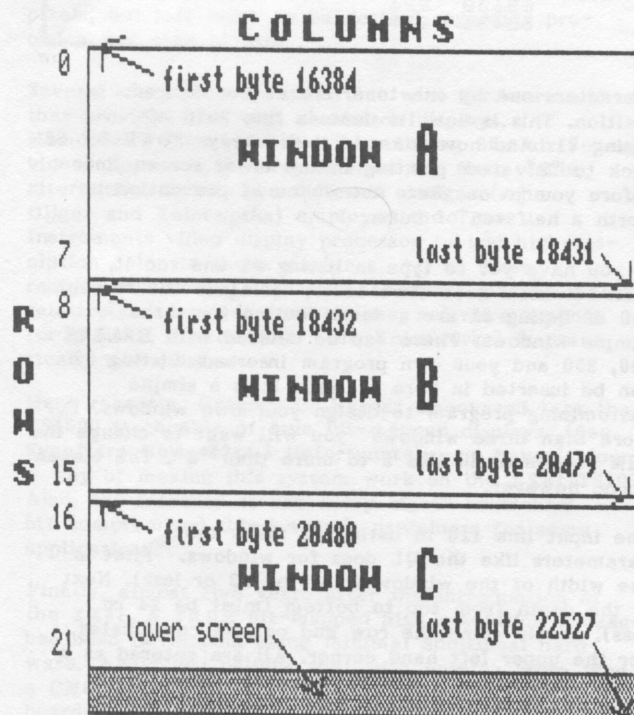
With the wave of new micros touting their window abilities maybe other 2068 owners like myself have been keeping quiet. Well, if you have been feeling sorry for yourself because you have a computer that "won't do windows" then cheer up! Those windows are painlessly easy on the mighty 2068!

Actually, the normal 24 row by 32 column screen is treated as three separate "windows" by the 2068 already. These begin as blocks in memory, each starting with their own address. These are shown as A, B, and C in figure 1. Try this little program line out after a CLS command: FOR t=16384 TO 22527: POKE t,255: NEXT t and watch the display. You should see a line form at the top of each character square successively until each square is filled in. Notice that the first block of eight rows completely fills in before even the first byte of the second block fills in. The display file is thus split into three blocks of 2048 bytes each.

Now no one reading this needs to have any more computing ability than knowing how to type in a program listing. I will try to explain as we go and let the 2068 do the rest. If you read Robert Hartung's article back in May/June '85 of SWN, he explained a nifty routine the Z80 chip in our 2068 possesses called LDIR. LDIR is great for moving huge chunks of memory (like the screen uses) around quickly. Line 5 of listing #1 among other chores sets up all the LDIR's machine code we will need.

Now to how we are going to do windows. With windows we want to temporarily overlay text or something (in the window) over what's there already. The hitch is not to lose what was there already! Windows aren't

Figure 1



much use if after we use them they leave a blank space. By reserving space enough to store the original screen contents, we can use the LDIR routine to put the original back on the screen unaltered. If our windows will only be appearing in one of the three blocks of the screen then reserving 2048 bytes for storage is ample. In listing #1 I use CLEAR 59198 in line 5 to give us enough room for all three at once -- the whole screen.

Listing 1.

```

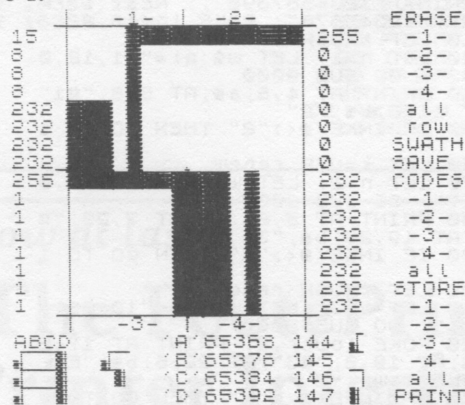
1 REM      2068 Windows!
5 CLEAR 59198: DIM w$(3,11):
LET a$="WINDOW": LET b$="press "
: LET store=59199: LET renew=592
11: LET lower=23659: FOR t=store
TO 59222: READ f: POKE t,f: NEX
T t: DATA 1,0,24,17,87,231,33,0,
64,237,176,201,1,0,24,17,0,64,33
,87,231,237,176,201: REM (sets u
P LDIR code)
10 FOR t=65368 TO 65431: READ
a: POKE t,a: NEXT t: DATA 15,8,8
,8,232,232,232,232,255,0,0,0,0,0
,0,0,255,1,1,1,1,1,1,1,1,232,232,2
32,232,232,232,232,232,1,1,1,1,1,1
,1,1,1,232,232,232,232,239,224,255,2
55,255,0,0,0,255,0,255,255,255,1
,1,1,255,0,224,224,224,224
100 POKE lower,0: FOR f=1 TO 76
: PRINT "1234567890": NEXT f: P
RINT "12345678": POKE lower,2
110 LET k=USR store
120 LET n=1: LET w$(n)="11,12,0
2,03": GO SUB 9000
140 PRINT AT 4,6;a$;AT 6,8;"#1"
;AT 11,6;b$;"2"
150 IF INKEY$<>"2" THEN GO TO 1
50
160 LET k=USR renew
170 LET n=2: LET w$(n)="13,20,0
2,17": GO SUB 9000
180 PRINT AT 5,21;a$;AT 7,23;"#
2";AT 19,20;b$;"3"
190 IF INKEY$<>"3" THEN GO TO 1
90
200 LET k=USR renew
210 LET n=3: LET w$(n)="10,08,1
6,04": GO SUB 9000
220 POKE lower,0: PRINT AT 17,6
;a$;AT 19,8;"#3";AT 22,6;b$;"E":
POKE lower,2
230 IF INKEY$<>"e" THEN GO TO 2
30
240 CLS: PRINT AT 8,6;"": FOR
t=1 TO 20: READ a: PRINT CHR$ a
: NEXT t: DATA 77,111,118,101,3
2,111,118,101,114,32,77,65,67,10
5,110,116,111,115,104,33
250 GO TO 9999
9000 LET w=VAL w$(n, TO 2): LET
d=VAL w$(n,4 TO 5): LET r=VAL w$
(n,7 TO 8): LET c=VAL w$(n,10 TO
11)
9010 IF r+d>24 OR c+w>32 THEN PR
INT AT 21,10;" Window's too big!
": STOP
9020 IF r+d>22 THEN LET low=0: L
ET r3=24-r-d: GO TO 9040
9030 LET low=2: LET r3=2:
9040 LET rd=r+d-2: LET cw=c+w-3
9050 POKE lower,low: PRINT AT r,
c;CHR$ 144: FOR t=c TO cw: PRIN
T CHR$ 145: NEXT t: PRINT CHR$
146
9060 FOR t=r+1 TO rd: PRINT AT t
,c;CHR$ 147:
9070 FOR f=c TO cw: PRINT " ":
NEXT f: PRINT CHR$ 148: NEXT t
9080 IF r3=0 THEN OVER 1: FOR t=
1 TO 32-w: PRINT " ": NEXT t:
OVER 0: PRINT CHR$ 149: FOR t=c
TO cw: PRINT CHR$ 150: NEXT t:
PRINT CHR$ 151: POKE lower,2: RE
TURN
9090 PRINT AT rd+1,c;CHR$ 149:
FOR t=c TO cw: PRINT CHR$ 150:
NEXT t: PRINT CHR$ 151: POKE low
er,2: RETURN
9900 SAVE "windows" LINE 5

```

In drawing quality window outlines we could use the DRAW command to make lines and shadows. But using the 2068 UDGs is quicker and cleaner. Line 10 sets up all eight UDG characters we will need for elegant windows. This particular line abandons the need for BIN statements as suggested in the User Manual. (This is the technique I used in the Tyd/bit column of the Jan/Feb '85 SWN.) Once set up these UDG characters can be printed as easily as any symbol. These graphic characters are detailed in figures 2 and 3. The BASIC in listing #1 lines 9000 onward calculate where each UDG goes in order to form nice window borders.

The routines in lines 9020 and 9080 deal with the lower two lines of the screen. The User manual refers to these as "not usable". Its in these lines that the 2068 badgers us for all the foolish syntax errors we made in our programming. Well, these lines are not at all off limits if we obey a few simple rules. If we POKE a zero into the system variable DF SZ at 23659 (also a no-no in the User Manual--see pg 263) the 2068 is lead to believe it has zero lines at the bottom of the screen. PRINT AT 22,0 works! PRINT AT 23,0 does not work, though. (Unless you wanted a good crash.) AT, TAB, and commas don't apply to line 23. The only way is to start in line 22 and print spaces over the existing screen

Figure 2.



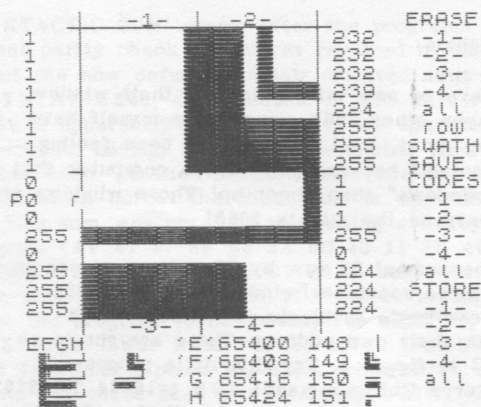
CURRENT U.D.G. CONTENTS:

```

CHR# 144 'A'
65368 15
65369 0
65370 0
65371 0
65372 232
65373 232
65374 232
65375 232
CHR# 145 'B'
65376 255
65377 0
65378 0
65379 0
65380 0
65381 0
65382 0
65383 0
CHR# 146 'C'
65384 255
65385 1
65386 1
65387 1
65388 1
65389 1
65390 1
65391 1
CHR# 147 'D'
65392 232
65393 232
65394 232
65395 232
65396 232
65397 232
65398 232
65399 232

```

Figure 3.



CURRENT U.D.G. CONTENTS:

```

CHR# 148 'E'
65400 1
65401 1
65402 1
65403 1
65404 1
65405 1
65406 1
65407 1
CHR# 149 'F'
65408 232
65409 232
65410 232
65411 232
65412 224
65413 255
65414 255
65415 255
CHR# 150 'G'
65416 0
65417 0
65418 0
65419 255
65420 0
65421 255
65422 255
65423 255
CHR# 151 'H'
65424 1
65425 1
65426 1
65427 255
65428 0
65429 224
65430 224
65431 224

```

characters one by one to arrive at the desired position. This is how its done in line 9080 of listing #1. And now for rule two: always POKE DF SZ back to "2" after printing in the lower screen lines before you go on. Here the ounce of prevention is worth a half ton of cure.

If you have yet to type in listing #1 and run it, then now is a good time to try it. Lines 100 through 250 of listing #1 are a demo section for three sample windows. These can be deleted with DELETE 100, 250 and your own program inserted. Listing #2 can be inserted in here as well. It is a simple customizing program to design your own windows. For more than three windows you will want to change the DIM statement in line 5 to more than "3". The eleven stays, however.

The input line 110 in listing #2 takes the parameters like the QL does for windows. First is the width of the window (must be 32 or less). Next is the depth from top to bottom (must be 24 or less). Lastly come the row and column coordinates for the upper left hand corner. All are entered as a string with commas in between as follows: xx,xx,xx,xx. If a number is only one digit, say 2, then type in 02. Lines 120, 170 and 210 of listing #1 are good examples of this. In listing #1 these

lines serve to set up the number of the window (n) and the window string (w\$) in lieu of listing #2's INPUT statement. After w\$(n) is set with the right parameters a simple GOSUB 9000 does the rest! Any text we want in the window can be done in BASIC as usual such as in listing #1's lines 140, 180, and 220. Line 220 prints into the lower screen and also POKEs DF SZ with zero and then 2 as we discussed previously. Can't remember where DF SZ is? Not to worry--I set a variable "lower" equal to the address.

I don't like to keep track of call addresses, either, so two more variables are set and ready from line 5: "store" and "renew". All we need to do is make a user call such as LET k=USR store to run the LDIR machine code that stores the screen safely away. Likewise LET k=USR renew puts the stored

Listing 2.

```
100 INPUT "window #=" ;n: INPUT
"WINDOW w,d,r,c ";w$(n)
110 POKE lower,0: FOR f=1 TO 76
: PRINT "1234567890";: NEXT f: P
RINT "12345678": POKE lower,2
120 LET k=USR store
130 GO SUB 9000
140 IF INKEY$="" THEN GO TO 140
150 LET k=USR renew: STOP
```

original back on the screen. See lines 120 and 150 of listing #2 for examples.

We have pretty well walked through both entire programs. There is a loop in line 100 of listing #1 that simply fills the screen, some key checks in lines 150, 190 and 230, and a sign-off message in line 240. None of these is essential. I am hoping that this little 1.7K program will be a good springboard for inventiveness. Machine code for the renewing of the screen is blinding fast. I intend to whittle back on the cumbersome BASIC slowness of the window drawing with machine code, too, in the future. If anyone else desires tackling improvements in this area I might suggest some good SWN reading: Ray Kingsley's ROM calls in Sept/Oct '84, P.S. Skipper's 2068 Messages in Mar/Apr '86, and William Power's M.C. Print Statement in May/Jun '85.

If any of you have suggestions or comments, I would love to hear from you! Also, if you are not up to typing in the long listings here, or would like a copy (on a new cassette with documentation) of the UDG designer program that produced figures 2 and 3 just send \$5 (or your own cassette and \$3) to: Paul Bingham, P.O. Box 2034, Mesa, AZ 85204.

WRX16 HI-RES FOR THE 1000

By Wilf Rigter

Introduction

For several years, ZX81/TS1000 owners have dreamed of a simple bit-mapped high-resolution system for their machines. The quest began with the appearance of "software-only" high-resolution packages, which offered an improvement over the crude 44x64 block pixels, but left much to be desired regarding precision and ease of use.

Several manufacturers came out with hardware add-ons that provided true high resolution displays. Memotech's HRG modified the existing Sinclair "cheap video" approach, and represented the first viable alternative. Later, two different designs (John Oliger and Kolorworks) employed the Texas Instruments video display processor to add high-resolution color to these machines. All of these require substantial additional hardware, making them relatively expensive. Furthermore, software support for at least one of them (the Kolorworks) is practically non-existent.

More recently, Gregory C. Harder discovered that the TS1500 is capable of true bit-mapped displays. (See SyncWare News #3:4.) Unfortunately, we haven't found a way of making this system work on the ZX81/TS1000. Also, the inability to set every eighth horizontal bit compromised this system's usefulness for some applications.

Finally, almost five years after the introduction of the ZX81, a TRUE bit-mapped high resolution package has been developed, using minimal additional hardware. Would you believe that with nothing more than a CMOS memory board (such as a modified "Hunter" board) you can now have 256x192 bit-mapped displays? Believe it! WRX16 HI-RES is here! (WR=Wilf Rigter, "x16" is the degree of resolution improvement.)

WRX16 HI-RES was developed by Wilf Rigter with the ZED Team, a sub-group of the Vancouver TS User Group, consisting of Mr. Rigter, Dave Ross and James Horne. It has been extensively tested; in fact, if you bought "SINC-ARTIST HR" from Weymil Corp., you've already seen the spectacular displays that this system can give.

Before we get on with the details, here is a summary of the advantages of this remarkable software:

- * 256x176 resolution, expandable to 256x206 (26 lines). ALL bits may be set/reset.
- * A single POKE instantly reverses any of the 32 columns; a short machine-code routine instantly reverses the entire screen. Software-only video reverse!
- * The high-res display file does NOT require stop bytes at the end of each display line, as does the standard Sinclair video or any of the other software-only hi-res packages. This considerably simplifies bit computations. Furthermore, you CANNOT crash the system by POKEing illegal codes into the display file.
- * Easily adapted to fill entire screen (to 26 rows), or leave the bottom two rows available for standard video (allowing INPUT, report codes, and BASIC command or program line entry).

Suddenly, the door is open to all the projects we have so far only dreamed about; 40+ columns, "what you see is what you get" word processing, CAD, \$100 "MAC-alikes," and on and on.

The key to the system is a succinct 76-byte algorithm, presented in this article, which utilizes the built-in display and memory refresh facilities in a novel way. This routine creates the hi-res screen, which can be POKEd, scrolled, inverted or blanked out with all the ease of a true bit-mapped display.

Copyright Notice

Note that this article and the WRX16 HI-RES core routine are the copyrighted property of Wilf Rigter. Except for your own use, please do not reproduce it in any form without written permission. If you plan to use this core in software for sale or other distribution, contact the author for licensing arrangements. Write to Wilf Rigter, c/o D. Ross Educational Electronics, 208-206 East 6th, Vancouver BC Canada.

[I should add that Mr. Rigter and the ZED Team are presently polishing up other highly significant developments for the ZX81. Whether or not these will be released for publication in this magazine depends entirely on the integrity of our readers. 'Nuff said? -fn]

Hardware Requirements

As stated, the hardware requirements for running WRX16-HR are minimal. Most of you already have what it takes. The CPU must be able to activate all 8 low address lines during the refresh cycle, instead of the conventional 7, and 8K of static RAM is required to store the hi-res display file.

Although most TS1000's have been supplied with a suitable CPU, some of the ones we have encountered have not. Suitable CPU's are, however, readily available at low cost. You'll know that you need a different Z80A or Z80B if every eighth horizontal bit is missing.

The 8K static RAM is mapped in the 8-16K address space for simplicity and convenience. Note that this must be STATIC RAM; dynamic RAM (such as used by 64K RAM packs) will not work. This is because this hi-res system relies on being able to fetch the video data during refresh time. On dynamic RAMs this is not possible; you must therefore arrange for 8K of static RAM in the 8-16K region.

The NVM circuit published in the last issue of SWN will work with no changes whatever. So if you've put off building this, here's a good reason to get out your soldering iron and go to it.

Another possibility is the ubiquitous "Hunter" NVM board. However, you will have to make a slight modification to make this work properly. The Hunter board uses the RD NOT signal to control the OE NOT pin of the CMOS RAM chips. As a result, the contents of RAM are gated onto the data bus only when RD NOT goes active low. However, during refresh time, the CPU holds RD NOT high, making it necessary to OR the RD NOT and RFSH NOT signals.

Such an OR circuit can consist of nothing more than a 1N4148 diode and a 4.7K resistor. Cut the trace connecting RD NOT (edge connector pin 16A) to the RAM pins 20. Bridge this gap with the 4.7K resistor. Connect the diode with its cathode towards RFSH NOT (edge connector pin 23A), and the anode to the RAM OE NOT pins (pin 20; the RAM end of the new resistor).

The only known software which is adversely affected by this modification is TS1500 HI*RES and packages (such as DUNGEON OF YMIR V1) which utilize this. If

you plan to run this type of software, install a switch in series with the diode so that the modification can be switched out.

Also see the "Support" section of this article for information on static RAM parts kits, if you want to "roll your own" but don't care to hassle with parts acquisition.

Finally, a functional equivalent to the "Hunter" board, with additional features, is being jointly developed by Larken Electronics and Silicon Mountain Computers. This will support both types of hi-res software with no changes. Stay tuned to SWN "For Your Support" for availability.

The only other hardware requirement is that the 48-64K region NOT be decoded. This precludes the use of 64K RAM packs, since this hi-res system relies on the 48-64K "echo" of the 16-32K region. It is conceivable that this restriction will be overcome, but at this writing a satisfactory fix has not been worked out.

And that's it! Nothing more is required, except of course the all-important software. In fact, the principles of the system can be demonstrated with a completely stock, 2K TS1000, albeit with reduced display size, minimal user friendliness, and some fiddling with the software.

TS1500: These routines will only work on the TS1500 if a 16K RAM pack is installed in addition to the CMOS RAM. Why this is, is unclear at this writing.

The Core Routines

Once you have your hardware properly set up, enter the core routine. Start with a 1 REM statement, 128 bytes long. (Type 1 REM followed by 128 X's or other character.) While you're at it, enter a line 2 REM of 104 bytes, and a line 3 REM of 148 bytes, which we'll fill later. Then enter the loader program of (lines 9000-9991 of Listing 4), and RUN 9000. Enter 16514 for "START" and 128 for "BYTES." Now enter the decimal values of Table 1, or use an assembler or Hot Z to enter the hexcode / mnemonics of Listing 1.

TABLE 1: WRX16 CORE (DECIMAL)

```
START? 16514    BYTES? 128
000:237:079:000:000:000:000:000:
000:000:000:000:000:000:000:000:
000:000:000:000:000:000:000:000:
000:000:000:000:000:000:000:000:
000:000:000:195:191:064:243:062:
007:071:000:016:253:198:238:060:
032:253:006:176:017:032:000:033:
000:032:195:198:064:000:000:000:
202:205:064:025:124:237:071:125:
195:131:192:042:012:064:042:012:
064:042:012:064:000:017:247:130:
025:062:030:237:071:062:245:001:
007:002:205:181:002:205:146:002:
205:032:002:221:033:168:064:195:
164:002:221:033:168:064:201:062:
030:237:071:221:033:129:002:201:
```


LISTING 1: WRX16 CORE

4082 00	IREM NOP		
4083 ED4F	DUMY LD R,A	:Refresh Counter LSB	
4085 00	NOP	:Collapsed D_FILE	
4086 00	NOP	(32 locations)	
4087 00	NOP	Executed each	
4088 00	NOP	HLINE, controls	
4089 00	NOP	32 columns.	
408A 00	NOP	NORM=00,	
408B 00	NOP	INVERT=80	
408C 00	NOP		
408D 00	NOP		
408E 00	NOP		
408F 00	NOP		
409A 00	NOP		
409B 00	NOP		
409C 00	NOP		
409D 00	NOP		
409E 00	NOP		
409F 00	NOP		
40A0 00	NOP		
40A1 00	NOP		
40A2 00	NOP		
40A3 00	NOP		
40A4 00	NOP		
40A5 C3BF40	JP DP-1	:Jump to Display Loop	
40A8 F3	DPLY DI	:PART 1: Disable Int.	
40A9 3E07	LD A,07	:Horiz. sync delay to	
40AB 47	LD B,A	:Center screen	
40AC 00	DP-0 NOP	:(coarse)	
40AD 10FD	DJNZ DP-0		
40AF C6EE	ADD A,EE	:(fine)	
40B1 3C	DP-A INC A		
40B2 20FD	JR NZ DP-A	:Set HIRES scrn size	
40B4 06B0	LD B,B0	:Horiz line length	
40B6 112000	LD DE,0020	:HIRES screen location	
40B9 210020	LD HL,2000	:Jump to display loop	
40BC C3C640	JP DP-2	:PART 2	
40BF 00	DP-1 NOP	:Delay 8 T-states	
40C0 00	NOP	:Dec. line counter	
40C1 05	DEC B	:If finished, jump.	
40C2 CACD40	JP Z DP-3	:Else calc. next line	
40C5 19	ADD HL,DE	:Load MSB of next	
40C6 7C	DP-2 LD A,H	line start addr.	
40C7 ED47	LD I,A	:Get LSB ready	
40C9 7D	LD A,L	:PART 3: Jump to ECHO	
40CA C383C0	JP C083	:PART 4	
40CD 2A0C40	DP-3 LD HL,(DFIL)	:D_FILE pointer	
40D0 2A0C40	LD HL,(DFIL)	:Delay 36	
40D3 2A0C40	LD HL,(DFIL)	T-states	
40D6 00	NOP		
40D7 11F782	LD DE,82F7	:Last line offset	
40DA 19	ADD HL,DE	:Last line address	
40DB 3E1E	LD A,1E	:Restore Sinc O/S	
40DD ED47	LD I,A	:font pointer	
40DF 3EF5	LD A,F5	:Sinc O/S needs A=F5	
40E1 010702	LD BC,0207	:1 row/8 high	
40E4 CDB502	CALL 02B5	:Call ROM	
40E7 CD9202	CALL 0292	:routines to display	
40EA CD2002	CALL 0220	:the last rows	
40ED DD21A840	LD IX,DPLY	:Point back, then	
40F1 C3A402	JP 02A4	:Jump to Sinc O/S	
40F4 DD21A840	HRES LD IX,DPLY	:Load JP vector	
40F8 C9	RET	:to HIRES routine	
40F9 3E1E	NRML LD A,1E	:Load font table ptr.	
40FB ED47	LD I,A	:and set JP vector	
40FD DD218102	LD IX,ZXIX	:to point to SOS	
4101 C9	RET	:display routine.	
4102 76	HALT		

Theory

This section is not essential reading if you only want to experiment with the WRX16 HI-RES core. However, we recommend that you do read it through at least once. Even if you don't understand it all at first, you will probably pick up quite a bit of information about this unique type of hi-res specifically, and about "cheap video" in general.

Turn It On

Entry from the Sinclair Operating System (SOS) is made by first calling "HRES", a short routine located at 40F4h. There, the IX register is loaded with a jump vector pointing to "DPLY", the entry point located at 40A8h.

When the SOS next encounters the JP (IX) instruction, program control is transferred to the HIRES routine instead of the normal one residing in ROM. This occurs 60 times a second during the NMI service routine.

Main Loop

Part 1 of the Hi-Res routine initiates the main display loop. The maskable interrupt is disabled, the hi-res screen is centered, its vertical and horizontal size is set, and HL is loaded as a pointer to the start address of a 6K memory block reserved for the hi-res display file.

Parts 2 and 3 make up the main display loop. Timing is critical because the routine must trigger the ULA

GREAT BOOKS FOR YOUR SINCLAIR

OFFICIAL 2068 TECHNICAL MANUAL.....\$25.00

CONTROL THINGS with your
Timex/Sinclair by R.L. Swarts.....\$5.00

ZX81: Programming for Real Applications
by Randle Hurley.....\$6.00

Both "CONTROL THINGS" and "ZX81".....\$10.00

Build a Microcomputer - Controlled NEW!
Robot by B.C. Taylor.....\$17.00

The Best of TIME DESIGNS Volume 1.....\$12.00

Taking the Quantum Leap Available February 1987
by Mike de Sosa.....\$26.00

WE PAY THE POSTAGE
ON ALL ORDERS!



Note: 4% charge added to all Credit Card Orders.

Send Name, Address, along with check/MO OR Credit Card No./Exp. date to:
TIME DESIGNS MAGAZINE COMPANY
29722 Hull Rd., Colton, OR 97017 (503) 824-2658

at precise intervals so that it, in turn, will produce line synchronization signals at appropriate intervals.

Part 2 keeps track of how many lines are left to do. If the display has been completed, a jump is made to Part 4, the Exit routine. Otherwise, the various registers are updated. The line start pointer is incremented by 32d, and the line count register is decremented by one. Finally, the start address of the next display line is loaded into the I and R registers.

In Part 3, a jump is made to the high-memory echo of a 32 character dummy display file, DUMY. There, the display facilities of the ULA are triggered and utilized as explained below. Refer to Mike Lord's "The Explorer's Guide..." for background on how the ULA operates.

- 1: The ULA is triggered into the display mode whenever address line A15 is high during an instruction fetch cycle (M1). The HI-RES software achieves this by way of a jump vector pointing to ECHO, the high memory phantom of DUMY. This method is similar to the one used in the SOS display routine.
- 2: On being triggered, the hardware swings into action. It generates a line sync pulse, and pulls the data lines low on each M1 cycle. The microprocessor "sees" the 00h NOP code, so for 4 T-states does nothing except feed the R and I registers' contents onto the address lines during the refresh cycle. It then increments the R register, and fetches the next dummy instruction. This process is repeated to the end of DUMY, where a jump instruction transfers control back to Part 2, completing the loop. This jump is made to its true address, so line A15 will no longer go high, temporarily suspending the ULA's display activities. This process is repeated until all lines have been displayed.
- 3: When in the display mode, the ULA forces levels onto the address lines which normally address the font tables resident in ROM. Since the HI-RES screen is mapped elsewhere in memory, this action has no effect.

Instead, with the contents of the I and R registers applied to the address lines, memory locations in the HI-RES display file are accessed, and their contents applied to the data bus.

- 4: During the refresh cycle, the ULA loads these bytes into an internal register. From there they are sent serially to the modulator for display.
- 5: During the M1 cycle, the ULA monitors data line D7 and if high, inverts the video data. For a normal display, characters of the dummy display file are set to 00h. When set to 80h, they serve to invert the display column by column.

As mentioned above, Part 4 of the routine is to return to the SOS after completing some house-keeping chores.

S.O.S. Return

To restore the normal display facilities, "NRML" (a short routine located at 40F9h) is called to restore the IX register to point to the usual ROM-based SOS display routine. The I register is also restored so that it points to the proper font table base address in ROM.

Modifications

As shown, the core routine gives 176 HI-RES lines, leaving the bottom two rows (16 lines) open for the usual error reports and INPUT lines. To change the number of HI-RES lines to, say 192, change the command at 40B4h to LD B,C0. Note that this will still keep the two input lines at the bottom, for a total of 26 lines. This might cause some TV's or monitors to spill off the top or bottom, and furthermore slows down machine operation slightly because of the additional lines displayed during the interrupt.

If you don't need the standard-video input rows at the bottom, replace the CALL 02B5 at 40E4h with NOPs (00). In principle, you could increase the number of HI-RES lines to 206, with LD B,D0 at 40B4h.

Making these or other changes might mess up the timing enough to visibly displace the HI-RES screen. You can trim this with the commands at 40A9h ("coarse") and 40AFh ("fine").

Some Service Routines

Now that you have the core routine installed, how can you use it? In principle, at least, anything you could wish to do could be accomplished using BASIC. For example, to reverse a single column, POKE (16517+N),128 where N is the column 0-31. To restore it to normal, POKE it back to zero. Similarly, you can POKE anywhere in the HI-RES display file to fill any desired byte (row of 8 pixels) with the desired pattern.

Other operations however, such as clearing the screen, printing and plotting, would be painfully slow in BASIC. Such jobs require machine-code routines to be useful. Listing 2 shows a few rudimentary routines (by fn) to CLS, PLOT, UNPLOT and REVERSE. Use the loader (START=16648, BYTES=104) to enter the values of Table 2 into your 2 REM statement. To use PLOT or UNPLOT, first POKE 16438 with the desired X value (0-255) and 16439 with the Y value (0-176, or higher if the core was suitably modified.) Though this is a very "bare-bones" operating system, it is capable of neat displays, as Paul Bingham's "CRATER" program in Listing 4 demonstrates.

TABLE 2: SERVICE ROUTINES (DEC.)

```
-----
START? 16648      BYTES? 104
033:000:032:058:181:064:031:031:
031:071:014:000:054:000:011:084:
093:019:237:176:201:033:133:064:
001:032:000:126:198:128:119:024:
237:253:203:059:214:024:004:253:
203:059:150:237:075:054:064:058:
181:064:061:144:218:173:014:087:
089:203:058:203:027:203:058:203:
027:203:058:203:027:033:000:032:
025:086:121:230:007:060:001:000:
008:203:002:203:017:061:032:012:
253:203:059:086:040:004:203:193:
024:002:203:129:016:235:113:201:
```


LISTING 2: SERVICE ROUTINES

4108 210020	CLS-°LD HL,2000	:Start of HR D_File	413F 57	GADR LD D,A	:Get address, Y val. in D
410B 3AB540	LD A,(40B5)	:# of lines?	4140 59	LD E,C	:X val. in E
410E 1F	RRA	:Compute # of bytes	4141 CB3A	SRL D	:Then divide
410F 1F	RRA	by mult. A	4143 CB1B	RR E	:by 8
4110 1F	RRA	by 32,	4145 CB3A	SRL D	to get
4111 47	LD B,A	and transfer	4147 CB1B	RR E	offset
4112 0E00	LD C,00	to BC.	4149 CB3A	SRL D	in DE
4114 3600	LD (HL),00	:Clear first byte	414B CB1B	RR E	
4116 0B	DEC BC	:One less to fill	414D 210020	LD HL,2000	:get base address
4117 54	LD D,H	:Tranfer	4150 19	ADD HL,DE	:and add offset.
4118 5D	LD E,L	to DE	4151 56	LD D,(HL)	:Get byte at addr.
4119 13	INC DE	:Next location	4152 79	LD A,C	:Which bit? get X val.
411A EDB0	LDIR	:Fill entire D_File	4153 E607	AND 07	:reduce mod 8
411C C9	RET	:then return.	4155 3C	INC A	:and increment
411D 218540	RVRSLD HL,4085	:Start of dummy DF	4156 010008	LD BC,0800	:B=8 bits, clear C
4120 012000	LD BC,0020	:# of bytes	4159 CB02	BTLPLC D	:rotate byte
4123 7E	LD A,(HL)	:get first byte	415B CB11	RL C	:into C
4124 C680	ADD A,80	:complement bit 7	415D 3D	DEC A	:Count down bit
4126 77	LD (HL),A	:and put it back	415E 200C	JR NZ NXBT	:until desired bit reached.
4127 18ED	JR 4116	:then fill the rest.	4160 FDCB3B56	BIT 2,(CDFG)	:Check flag
4129 FDCB3BD6	PLOTSET 2,(CDFG)	:Set flag for PLOT	4164 2804	JR Z RSBT	:for Plot/Unplot
412D 1804	JR PLT*	:Go to PLT* routine	4166 CBC1	STBTSET 0,C	:Set the bit
412F FDCB3B96	UNPLRES 2,(CDFG)	:Set flag for UNPLOT	4168 1802	JR NXBT	:and keep rotating
4133 ED4B3640	PLT*LD BC,(COORD)	:Get coords. in BC	416A CB81	RSBTRES 0,C	:or reset bit
4137 3AB540	PLBCLD A,(40B5)	:How many lines?	416C 10EB	NXBTDJNZ BTLPLD (HL),C	:and keep rotating
413A 3D	DEC A	:Max. vert. = 1 less	416E 71	RET	:Replace modified byte
413B 90	SUB B	:Invert vert. axis	416F C9		:and return.
413C DAAD0E	JP C 0EAD	:Overrange - error.			

The last routine (Listing 3) is a machine-code "Mandala" program to demonstrate the kind of speed that can be achieved using machine code PLOT and UNPLOT. Use the loader (START=16758, BYTES=148) to input the values of Table 3 into your 3 REM statement.

Finally, enter the rest of the BASIC DEMO lines of listing 4. Run the program. Briefly touching a key during the Mandala portion reverses the screen. Press BREAK to exit, then CONT to run the Crater portion. It pauses after every line so you can see how it's progressing. Pretty neat, eh what?

Support

Support for WRX16 HI-RES is available from D. Ross Educational Electronics, [see address in the "Copyright" section, or call (604) 874-1756]. The ZED Group and D.R.E.E. are preparing RESPACK 1 and

RESPACK 2 starter-kits to make it easy for non-hackers to get going. Dave Ross has been involved with development from the start, and intends to stock the required kits, and later assembled units with additional refinements including bank-switched memory. Also contact D.R.E.E. if you develop interesting applications using this system.

Conclusion

We believe that the WRX16 HI-RES system removes the final obstacle to a more widespread acceptance of the TS1000. Dave, Jim Horne and myself are committed to developing this to the fullest, and will continue to work on user software and hardware enhancements. "In the wings" are at least one comprehensive Extended BASIC utility. Options will include easy plot, unplot and complement pixels; DRAW lines, circles, and other shapes; sprites; smooth (bit-

LISTING 3: HI-RES MANDALA

4176 CDF440	°CALL HRES	419C 67	LD H,A	41BE 1804	JR 41C4	41EE CD460F	CALL BREK
4179 CD0841	CALL CLS-	419D 7D	LD A,L	41C0 FDCB3BD6	SET 2,(CDFG)	41F1 D2F940	JP NC NRML
417C CDBB02	CALL KEYN	419E 90	SUB B	41C4 C5	PUSH BC	41F4 1886	JR 417C
417F 2C	INC L	419F 3001	JR NC 41A2	41C5 CDF641	CALL 41F6	41F6 3E80	LD A,80
4180 C41D41	CALL NZ RVRSLD DE,004B	41A1 25	DEC H	41C8 CD0042	CALL 4200	41F8 81	ADD A,C
4183 114B00	LD DE,004B	41A2 6F	LD L,A	41CB CD3741	CALL PLBC	41F9 4F	LD C,A
4186 ED4B3240	LD BC,(SEED)	41A3 2B	DEC HL	41CE C1	POP BC	41FA C9	RET
418A 03	INC BC	41A4 223240	LD (SEED),HL	41CF C5	PUSH BC	41FB 3E80	LD A,80
418B 62	LD H,D	41A7 CB3D	SRL L	41D0 CDF641	CALL 41F6	41FD 91	SUB C
418C 68	LD L,B	41A9 4D	LD C,L	41D3 CD0542	CALL 4205	41FE 18F9	JR 41F9
418D CD0913	CALL 1309	41AA 7C	LD A,H	41D6 CD3741	CALL PLBC	4200 3E58	LD A,58
4190 E5	PUSH HL	41AB 0655	LD B,55	41D9 C1	POP BC	4202 80	ADD A,B
4191 62	LD H,D	41AD B8	CP B	41DA C5	PUSH BC	4203 47	LD B,A
4192 69	LD L,C	41AE 3803	JR C 41B3	41DB CDFB41	CALL 41FB	4204 C9	RET
4193 CD0913	CALL 1309	41B0 90	SUB B	41DE CD0042	CALL 4200	4205 3E58	LD A,58
4196 C1	POP BC	41B1 18FA	JR 41AD	41E1 CD3741	CALL PLBC	4207 90	SUB B
4197 7C	LD A,H	41B3 47	LD B,A	41E4 C1	POP BC	4208 18F9	JR 4203
4198 81	ADD A,C	41B4 FDCB3B56	BIT 2,(CDFG)	41E5 CDFB41	°CALL 41FB		
4199 3001	JR NC 419C	41B8 2806	JR Z 41C0	41E8 CD0542	CALL 4205		
419B 04	INC B	41BA FDCB3B96	RES 2,(CDFG)	41EB CD3741	CALL PLBC		

wise) scrolling; upper/lower case, mixed text/graphics, and a form of windowing, to mention but a few possibilities. The goal is to provide a complete system that outperforms similar hardware-based schemes at a fraction of the price.

Acknowledgements also to Ray Lanoville for assistance in preparing this article from my field notes. Ray is interested in the educational potentials of the TS1000, and will be contributing to the team effort in that area.

TABLE 3: HI-RES MANDALA (DEC.)

```

START? 16758      BYTES? 148
205:244:064:205:008:065:205:187:
002:044:196:029:065:017:075:000:
237:075:050:064:003:098:104:205:
009:019:229:098:105:205:009:019:
193:124:129:048:001:004:103:125:
144:048:001:037:111:043:034:050:
064:203:061:077:124:006:085:184:
056:003:144:024:250:071:253:203:
059:086:040:006:253:203:059:150:
024:004:253:203:059:214:197:205:
246:065:205:000:066:205:055:065:
193:197:205:246:065:205:005:066:
205:055:065:193:197:205:251:065:
205:000:066:205:055:065:193:205:
251:065:205:005:066:205:055:065:
205:070:015:210:249:064:024:134:
062:128:129:079:201:062:128:145:
024:249:062:088:128:071:201:062:
088:144:024:249:

```

LISTING 4: WRX16 HI-RES DEMO

```

5 LET HR=16628
7 LET NRML=16633
10 LET CLS=16648
20 LET RURS=16669
30 LET PLOT=16681
40 LET UNPL=16687
50 LET XP=16438
60 LET YP=16439
90 REM WRX16 HI-RES MANDALA
95 SLOW
100 RAND USR 16758
190 REM WRX16 HI-RES MANDALA
200 FAST
210 RAND USR CLS
215 IF PEEK 16517<>128 THEN RAN
D USR RURS
220 LET Q=-4
230 FOR A=23 TO -16 STEP -1.6
240 IF A>5 THEN LET Q=Q-1.7
250 IF A<5 THEN LET Q=Q+1
260 FOR B=Q TO 15 STEP .2
270 LET Z=COS (.05*(A*A+B*B))
290 LET H=80-A*.9-B-Z
300 LET X=(124-600*((-A*.4+B*.9
)/H))
310 IF X>255 OR X<0 THEN GOTO 3
70
320 LET Y=96+600*((-A*.3-B*.1+Z
*.7)/H)
330 IF Y>175 OR Y<0 THEN GOTO 3
70
340 POKE XP,X
350 POKE YP,Y
360 RAND USR PLOT
370 NEXT B
380 GOSUB 500
385 FAST
390 NEXT A
400 GOSUB 500
410 IF INKEY$="" THEN GOTO 410
420 RAND USR NRML
430 STOP
500 SLOW
510 RAND USR HR
520 FOR N=1 TO 50
530 NEXT N
540 RETURN
9000 REM WRX16 HI-RES MANDALA
9010 FAST
9020 CLS
9030 PRINT "START? ";
9040 INPUT ST
9050 PRINT ST,"BYTES? ";
9060 INPUT BY
9070 PRINT BY
9080 FOR N=ST TO ST+BY-1
9090 INPUT U
9100 POKE N,U
9110 PRINT (STR$ (U+1000)) (2 TO
);";";
9120 NEXT N
9990 SAVE "HRDEM"
9991 RUN

```

Now Available! "THE BEST OF SUM, PART II"

Over 60 pages of program listings, reviews, hardware projects, hints and tips, and articles. Covers TS-1000, TS-2068, and QL. All are reprints from the last year of **SUM** plus a few that didn't make it into print before now.

Articles include Building an EPROM Programmer, Sprites on the 2068, Adding RGB to 2068, QL Word Processing, What's Available for TS-1000, and much more. Does overlap the "Special Issues" advertised to the right.

\$6.95 postpaid

The original "THE BEST OF SUM" is now in its 2nd printing. 112 pages of articles going all the way back to the beginning of **SUM** — 3 years worth!

\$11.95 postpaid

QL Special Issue (12/85) — \$3.95 postpaid.

TS-1000 Special Issue (5/86) — \$3.95 postpaid.

Everything! \$20.00 postpaid

FOOTE SOFTWARE P. O. Box 14655 — Gainesville, FL 32604 904/462-1086 (6 pm - 9 pm EDT)

2068 VARIABLE SAVE TRICK

By Robert C. Fischer

Getting right to the point of this article, why would anyone want to save only the variables from a program? There are two reasons. First, some programs are used to store various sets of data. For example, I use PRO/FILE to keep track of student information, music lists, customers, and other things. I could not store all this at one time in the computer. If I had to load the program with the data in each case, I would waste considerable time. A second reason would be to use a different program to operate on the same data as another program originally created. Again, using PRO/FILE as an example, you may want to load your PRO/FILE data into another program which would do a search and replace operation or other functions. It's true that the 2068 permits you to save dimensioned data, but you may have other variables which serve as pointers that need transferring as well.

To save the variables without the program, you could DELETE 0,9999 before saving, but the SAVE and VERIFY commands would have to be entered directly. Also, you may still need to keep the program in the computer for further use. After a lot of experimenting, I may have a solution. The "may have" is because the process alters the system variables and, although it seems fine so far, I prefer to test it much more before using it with complete confidence.

The process simply involves fooling the computer about where to start saving. Since we only want the variables, we need to make the computer think that the start of the variables is the start of the entire program.

The computer stores things in certain places or in a certain order. For instance, the ROM is always found from address 0 to 16383. Next comes the display and attribute files followed by the printer buffer. Eventually we come to the start of the BASIC program which on the 2068 is normally found at address 26710, but it can be moved. Right after the BASIC ends, we find the start of the variables. The exact starting point varies according to the size of the BASIC program. Actually, when you first turn on the computer, the variables and the program start at the same place because no program exists.

The system variables help the computer keep track of these things. System variables are not like the variables we use (like LET X=10). Instead they are special storage addresses which tell the computer where to find things or to do certain things (like add a key click)

One of these variables is called PROG (short for program). This tells the computer where the program starts and you can find PROG stored in addresses 23635 and 23636. If you PEEK those addresses you will find the values 86 and 104. The actual start of the program is found by the formula $86+256*104$ which equals 26710. The start of the variables is similarly found by peeking VARS (addresses 23627 and 23628).

To start saving at the start of VARS, we first POKE PROG with the values presently in VARS. As long as no GOTO or GOSUB instructions are encountered before PROG is reset, the program will continue to function. After saving and verification is completed, PROG is set back to its original values. If an error occurs

```
5 PRINT AT 10,3;"PRESS A KEY
TO START TEST": PAUSE 0: CLEAR
10 LET A=1: LET B=2
11 LET A$="A": LET B$="B"
12 DIM X(2): LET X(1)=1000: LE
T X(2)=2000
13 DIM Z$(2,3): LET Z$(1)="ABC
": LET Z$(2)="DEF"
100 PRINT "A,B,A$,B$,X(1),X(2)
,Z$(1),Z$(2)
200 PRINT "PRESS TO CONTINUE
WITH SAVING": PAUSE 0: BEEP .1,3
0
1000 POKE 23635,PEEK 23627: POKE
23636,PEEK 23628
1010 SAVE "VARIABLES": PRINT "T
AB 8:"REWIND TO VERIFY": VERIFY
"VARIABLES"
2000 POKE 23635,86: POKE 23636,1
04
5000 CLEAR: PRINT "OLD VARIABLE
VALUES SAVED"/"HERE ARE NEW VAL
UES": LET A=-1: LET B=-2: LET A
$="a": LET B$="b": DIM X(2): LET
X(1)=-1000: LET X(2)=-2000: DIM
Z$(2,3): LET Z$(1)="abc": LET Z
$(2)="def": PRINT "A,B,A$,B$,X(1
),X(2),Z$(1),Z$(2)
5010 PRINT "REWIND TAPE TO LOA
D OLD VALUES": POKE 23635,PEEK 2
3627: POKE 23636,PEEK 23628: LOA
D "VARIABLES": POKE 23635,86: PO
KE 23636,104
5020 CLS: PRINT "RESTORED VARIA
BLES ="A,B,A$,B$,X(1),X(2),Z$(1
),Z$(2): STOP
9000 SAVE "TEST" LINE 1: PRINT A
T 10,0:"REWIND TO VERIFY PROGRAM
": VERIFY "TEST"
```

before it is reset, you will not find any BASIC until you poke the original values back. This same process works on the Spectrum except the original value of PROG is different. If you have Interface 1 connected to your Spectrum, you will find that PROG also changes as soon as SAVE, LOAD, VERIFY, or MERGE is used. Once it changes, it remains the same so you can still use this process. It will not work if saving to ZX Microdrives though, only tape. If you have any hardware or software installed which moves PROG, then you must adjust the values to reset PROG to use this program. To find the correct values, just set up your hardware or load your software and PEEK into PROG to get the correct numbers.

To load the saved variables into another program, first set PROG equal to VARS again and LOAD the variables, then reset PROG. At first I just MERGED the program without changing PROG and it worked on a small test program, but when trying on PRO/FILE I always got an out of memory error. By changing PROG it worked fine. Since I don't understand why this happens, I consider this experimental.

The accompanying listing shows a simple example for the 2068 and gives the changes necessary for the Spectrum with or without Interface 1.

Lines 5 through 200 first wait until you want to start, assigns different types of variables, prints them on the screen, and waits again to continue. The pauses also give you the opportunity to break in before PROG is changed if you wish. When I broke in on the second PAUSE during testing, the first POKE was sometimes activated, so I added the BEEP to prevent this.

Line 1000 starts the actual meat of the program and thus the part you would use in your own projects. This simply pokes PROG with the contents of VARS.

Line 1010 saves and verifies the program which only contains variables.

Line 2000 resets PROG to its original value. If you have a Spectrum, change the 86 and 104 to be 203 and 92. If Interface 1 is connected change the POKES to 5 and 93 respectively. Remember that Interface 1 does not change PROG until SAVE, LOAD, etc is activated, thus be sure this has been done first. This ends the actual SAVE process.

Line 5000 simply changes the values of the variables to prove we can later reload the original values. These new values are printed on screen.

Line 5010 first prints instructions. Next the actual LOAD section is activated: first PROG is poked with VARS, then the program of variables is loaded, then PROG is reset.

Line 5020 prints the restored variables which should be the same as when saved.

Line 9000 simply lets you make a copy of this test program.

I hope this works for you. If you have any questions, ideas, or discoveries, contact Robert C. Fischer; Rt 2 Arizona St; Emerson, Ga. 30137

BASIL'S COMPENDIUM

Basil Wentworth

Why Use Machine Code ?

A PHILOSOPHICAL DIGRESSION

After the lengthy Chapter 9, the brevity of this one may come as a welcome relief. The brevity exemplifies the purpose of this chapter, which might be summarized as advising you not to send a man to do a boy's job.

You might be tempted, after following me through 9 chapters, to say something like "Look, this seems to be an awfully complicated way to add 2+2. I can do the job faster in BASIC, or even with pencil and paper." I could not offer as a valid defense the fact that not all numbers are so easy to handle. BASIC would still do the job just as well. Sure, once we get the program written, the machine code will run in microseconds, compared with the milliseconds required by BASIC; but what does that matter, when the speed of your fingers and the response of your eye are measured in large fractions of a second?

For controlling machines, the jump from milliseconds to microseconds might justify the extra effort. The utility programs that have been given at the top of each chapter served as a reminder that there are things--many things--that machine code does better than BASIC can. But not everything. If your operations are too fast for the eye to follow, is the program improved by making them MUCH too fast for the eye to follow?

Of course, the miniature arithmetic programs we have presented so far were designed to illustrate principles, rather than to do something useful. ("What?" I hear you cry. "Isn't education useful in itself?" To which I reply, "Don't be a smart alec!") But here's a pseudo-practical case, and one that's not too different from some routines I've seen published in popular magazines.

I'm not going to give mnemonics for this program. You can work them out for yourself, if you want to.

Here's a hint that may save you a bit of time--the first 19 bytes are data. You might have guessed as much, from the fact that we USR our way into the program at 16533, rather than the usual 16514.

After you RUN the program, enter RAND (not PRINT) USR 16533. It's a rather roundabout way to write a love note, even if my wife did think it was rather cute.

10-2

```
100 LET P$="39.38.56.46.49.0.49
.52.59.42.56.0.47.52.40.42.49.62
.51."
101 LET P$=P$+"17.130.64.1.19.0
.205.107.11.201."
```

The point is, as I remarked at the head of this chapter, it's not always wise to send a man to do a boy's job. Don't become so enthralled with your ability to write machine code programs that you overlook some easier, or more reliable, way to do what you want. If you have the choice between simple and complicated, simple is usually the more elegant. It's often a poor bargain to trade many minutes of machine code writing for a saving of a few milliseconds in program execution.

But of course, if you want to do it just for fun...

There are jobs that machine code is best for. There are some jobs that machine code can do easily, but are hardly possible (or impossibly hard) for BASIC. Use your machine code for them. Usually these are tasks involving loops, or great numbers of repetitive calculations.

PRINT is often better done in BASIC. Loops are always faster in machine code. Arithmetic and trigonometric calculations tend to be easier in BASIC.

As a matter of general principle, you'll probably find it best to block diagram your programs in BASIC before you start writing machine code. Usually you'll get a feeling right there whether it's better to stick with BASIC or to use machine code.

And that's the end of a short, but important, chapter. There will be more.

WORDSEARCH

A Comparison between the ZX80 and TS1000

by Duncan Teague

Last year, during the return of the Comet Halley, my planetarium was planning to send a monthly newsletter to the local elementary schools. In order to provide a little diversion, a good word search puzzle was to be included in each newsletter. The program to generate such a puzzle was found in "THE BEST OF SYNC: VOL. I" (c)1983 Creative Computing Press, page 24).

A problem soon arose however, in that the program would work fine on the ZX80 for which it was written, but wouldn't work on a TS1000. After reading many sources comparing and explaining the differences between the two computers, several modifications were made to the original program in an attempt to get it to work on the TS1000. All to no avail.

The author of the original program, Bill McCray, was located and consulted. Mr. McCray stated that he had never produced a TS1000 version of his WORDSEARCH program. Furthermore, while several other people had contacted him about the problem of ZX80/TS1000 incompatibility, to date he was not aware of anyone successfully modifying his program to run on the TS1000. Mr. McCray did suggest that perhaps examining the sections of the program that depend on relational operators might shed some light on the incompatibility problem.

That was an area that had been overlooked during research of comparisons between the two computers. In fact, the differences as to how the two machines handle relational operators turned out to be the key to making the WORDSEARCH program operate successfully on the TS1000. In the process of translating from ZX80 Basic to TS1000 Basic, a number of "tricks" were discovered that allow successful conversion of programs between the two computers. Please follow along with the listing of the modified and now working program. Also note that in all references given below, ZX81 is equivalent to TS1000. Program lines have also been renumbered so they no longer match those given in "THE BEST OF SYNC" for the ZX80.

Before we delve into relational operators, a few other differences in ZX80 Basic vs. TS1000 Basic need to be explained. In lines 20 and 30 of the listing, two arrays are DIMensioned. The first holds the 165 characters of the 11 X 15 puzzle. The second holds the character codes for each letter of the 11 letter words that will be entered into the puzzle. On the two machines the lines would be entered differently:

```
ZX80: 20 DIM A(164)    ZX81: 20 DIM A(165)
      30 DIM B(10)      30 DIM B(11)
```

The differences arise because because a ZX80 array allows a 0th element. The array is therefore dimensioned 0 to 164.

Because a 0th element is possible, some of the FOR...NEXT loops in the program start and end with different variable values. The variable C, for instance, takes on the values 0 to 10 in the ZX80 version while it ranges from 1 to 11 in the ZX81 version.

```
ZX80: 170 FOR C=0 to 10
```

```
ZX81: 170 FOR C=1 to 11
```

It is also necessary to change lines 500 and 600 since the variable K takes on the values of C.

Line 200 contains a command to slice the first character of A\$ from the string after that character code has been stored in the B array. There is a satisfactory substitution for the ZX80 command TL\$:

```
ZX80: 200 LET A$=TL$(A$)
```

```
ZX81: 200 LET A$=A$(2 TO)
```

The random number generators of the two computers also operate differently. The ZX80 automatically makes an integer out of random numbers in the range of 0 to N and includes N in the range. The ZX81 does not. Line 300 shows how you must take the INT of the RND number on the ZX81 and add +1 to the result.

```
ZX80: 300 FOR I=1 to RND*8
```

```
ZX81: 300 FOR I=1 TO INT (RND*8)+1
```

The same consideration must be applied to lines 400 and 410.

Now back to the relational operators. When the ZX80 uses the relational operators =, >, and < to determine the equality or inequality, a result of false is evaluated as 0, and true is evaluated as -1. When the ZX81/TS1000 performs the same comparisons, false is still 0, but true is +1. Because of the differences in assigning a value for true, when converting from ZX80 Basic to ZX81 basic it's necessary to multiply some expressions containing relational operators by -1. Line 360 shows how this is accomplished. Note that the ZX81 version also has (C-1); that's because C is 1 greater in the ZX81 version than the ZX80 version.

```
ZX80: 360 LET D=-C*(U<0)
```

```
ZX81: 360 LET D=(C-1)*(U<0)
```

Once again, in the ZX81 line above, 1 has been subtracted from C since its value is one less than the length of the word to be embedded into the puzzle. C has also been multiplied by -1 to account for the differences in relational operators.

The same operations also hold true for lines 370, 380, and 390. They are not necessary for lines 910 and 920 since they involve the multiplication of two possibly negative numbers. The result will always be 0 or positive in these lines so no changes are necessary.

At this point the program functioned properly. One final communication from Bill McCray eliminated the annoying problem of words starting in the last column of the puzzle and wrapping around to the first. This change is related to the fact that the ZX80 array contained elements 0 to 164 instead of 1 to 165.

The variables M and P (lines 490 and 590, respectively) become subscripts of the A array (in lines 510 and 670). They should therefore take on values 1

to 165 in the ZX81 version of the program. To have them do so requires the addition of 1 to the lines in which the values of M and P were calculated.

```
ZX80: 490 LET M=X+15*Y
      590 LET P=X+15*Y
```

```
ZX81: 490 LET M=X+15*Y+1
      590 LET P=X+15*Y+1
```

Now the program works like a charm. You can keep entering words for as long as you like. Lengthy words should be entered first to give them a better chance of being properly embedded into the puzzle. If you're not happy with the way the program arranged the puzzle, reRUN the program.

If you try to enter a word longer than 11 characters or a word containing a non-alpha character, then you'll be asked to re-enter a different word. If there is no room left in the puzzle for a word you try to enter, try a shorter word. Program options allow you to display either the puzzle or its solution. The screen can be copied to your printer.

```
REM WRITTEN FOR THE ZX80 BY
  BILL MCCRAY
10 REM MODIFIED FOR TS/1000 BY
  DUNCAN R. TEAGUE
20 DIM A(165)
30 DIM B(11)
40 CLS
50 RAND
60 PRINT "  CREATS A WORDSEARCH
  PUZZLE"
70 PRINT
80 PRINT "WORDS MAXIMUM OF 11
  LETTERS EACH"
90 LET A$="FIRST"
100 PRINT
110 PRINT A$;" WORD?";TAB 15;"N
  ULL ENTRY TO END"
120 INPUT A$
130 CLS
140 IF A$="" THEN GOTO 720
150 PRINT A$
160 IF LEN A$>11 THEN GOTO 230
170 FOR C=1 TO 11
180 LET B(C)=CODE A$
190 IF B(C)<38 THEN GOTO 260
200 LET A$=A$(2 TO )
210 IF A$="" THEN GOTO 280
220 NEXT C
230 PRINT "TOO MANY LETTERS"
240 PRINT "RE-ENTER"
250 GOTO 120
260 PRINT "INVALID CHARACTER"
270 GOTO 240
280 LET U=1
290 LET V=0
300 FOR I=1 TO INT (RND*8)+1
310 GOSUB 910
320 NEXT I
330 LET A=-1
340 FOR I=1 TO 8
350 GOSUB 910
360 LET D=(C-1)*(U<0)
370 LET E=15-(C-1)*(U>0)
380 LET F=(C-1)*(U<0)
390 LET G=11-(C-1)*(U>0)
400 LET X=INT (RND*(E-D))+D
410 LET Y=INT (RND*(G-F))+F
420 FOR J=1 TO (E-D)*(G-F)
430 LET X=X+1
440 IF X>E THEN GOTO 480
450 LET X=0
460 LET Y=Y+1
470 IF Y>G THEN LET Y=F
480 LET B=0
490 LET M=X+15*Y+1
500 FOR K=1 TO C
510 LET L=A(M)
520 LET M=M+U
530 IF L=0 THEN GOTO 560
540 IF NOT L=B(K) THEN GOTO 510
550 LET B=B+1
560 NEXT K
570 IF NOT B>A THEN GOTO 510
580 LET A=B
590 LET P=X+15*Y+1
600 LET Q=U
610 NEXT J
620 IF A=-1 THEN GOTO 660
630 NEXT I
640 PRINT "WORD DOES NOT FIT"
650 GOTO 700
```

```
660 FOR K=1 TO C
670 LET A(P)=B(K)
680 LET P=P+Q
690 NEXT K
700 LET A$="NEXT"
710 GOTO 110
720 PRINT AT 21,0;"SOLUTION  B
  ZZLE COPY END"
730 INPUT A$
740 IF A$="E" THEN STOP
750 IF A$="C" THEN GOTO 960
760 CLS
770 LET I=1
780 FOR K=0 TO 10
790 FOR J=0 TO 14
800 LET L=A(I)
810 IF L THEN GOTO 840
820 LET L=27
830 IF NOT A$="S" THEN LET L=IN
  T (RND*26)+38
840 PRINT CHR$(L);" ";
850 LET I=I+1
860 NEXT J
870 PRINT
880 PRINT
890 NEXT K
900 GOTO 720
910 LET W=(U>-1)*(V>-1)-(U<1)*(
  V<1)
920 LET V=15*((U<1)*(V>-1)-(U>-
  1)*(V<1))
930 LET U=W
940 LET W=U+V
950 RETURN
960 PRINT AT 21,0;"..
970 COPY
980 GOTO 720
990 SAVE "WORDSEARCH"
1000 RUN
```

This is a bare bones program. Remember that the original program had to fit into the small memory of the ZX80. The user can easily add more programing lines to the ZX81 version to allow editing and printing out of the word list. Simple changes to the range of character codes will produce a TS2068 version of the WORDSEARCH program. Be sure to run the TS1000 version in FAST mode.

Included is a sample WORDSEARCH puzzle related to SyncWare News and its staff. Hidden in the puzzle are the following words: SYNCWARE, NEWS, TIMEX, COMPUTERS, SOFTWARE, HARDWARE, NACHBAUR, WOODS, BENT, and MOORE. See if you can solve the puzzle without peeking at the solution found elsewhere in this issue. GOOD LUCK!

Fun Corner

```
W H N Q I R T R C E N P L W E
F Y A T Z S Y N C W A R E B K
J Y M R E B P C E X C Z Q Z N
O L Q R D V U P C B H G U Z G
Q S O F T W A R E A B H J W U
A O D N H N A X V U A J Y R Y
M T I M E X S R E T U P M O C
F Q H K Y W J D E R R W I V Y
I Q I E E A S V O Y S Q W U X
T R L B G Y Q P I O R G W E S
K S Q K R D P B B Q W T H O N
```


2068 WORD WRAP UTILITY

Charles Stelding

When printing a text to the screen by using the PRINT statement, words at the end of the screen often will be cut off in the middle and continued on the next line. This makes the text difficult to read.

This utility will put the entire word at the beginning of the next line so that the word remains intact. You simply put the text you wish to print to the screen or a printer in a REM statment. A PRINT statement is not necessary. To illustrate how this works, type in the following BASIC listing and RUN it. Further explanation is given in the listing.

By changing the DATA STATEMENT in LINE 10, this Utility will also work on the Spectrum Computer. Change LINE 10 to read:

10 DATA "3E02CD01163E0DD72A5D"

```

9 REM For the SPECTRUM, change
  e line 10 to DATA "3E02CD01163E0
  DD72A5D"
10 DATA "3E02CD01163E0DD72A5D"
11 DATA "5C2323E51120001953D5"
12 DATA "7EFE202015C1E17EFE0D"
13 DATA "C8D72310F83E0DD77EFE"
14 DATA "2020E22318F8C12B057E"
15 DATA "FE0D28E4FE2020F518DE"
16 RESTORE 10: LET addr=65000:
  FOR J=10 TO 15: READ a$: FOR K=
  1 TO LEN a$: STEP 2: LET num=(COD
  E a$(K)-48-(7 AND a$(K)>"9"))*16
  +(CODE a$(K+1)-48-(7 AND a$(K+1)
  >"9")): POKE addr,num: LET addr=
  addr+1: NEXT K: NEXT J
100 LET wrap=65000
103 POKE (wrap+1),2: REM print
  to screen. POKE (wrap+1),3 will
  send text in the REM statement
  to printer. If you are using a
  printer other than the 2040, you
  must have your printer software
  loaded and initialized.
105 LET width=32: POKE (wrap+15
  ),width
110 RANDOMIZE USR wrap: REM T
  he REM statement must immediatel
  y follow the "RANDOMIZE USER wra
  p" entry as this example illustr
  ates. The letters following the
  REM token will be printed to th
  e screen or printer and words wh
  ich would normally be cut off in
  the middle at the end of the sc
  reen will be "wrapped around" to
  the next line to make them easi
  er to read.
120 RANDOMIZE USR wrap: REM A
  quotation mark is simply entere
  d as (") instead of two quotes (
  ") as in a PRINT statement.
130 RANDOMIZE USR wrap: REM T
  he code is saved by SAVE "REM WR
  AP"CODE 65000,60 and is relocata
  ble anywhere in useable memory.
  It will work on both the Spectr
  um and 2068 (also with the OS 64
  cartridge). To relocate the co
  de, simply define "addr" the val
  ue of the new location as "LET a
  ddr=32000" in line 16. Then def
  ine "wrap" as 32000 in line 100.
140 RANDOMIZE USR wrap: REM T
  he column width is now set at 32
  but can be changed to 64 for th
  e OS 64 cartridge. To change th
  e column width, type in any new
  width in line 105.
9998 STOP
9999 SAVE "REM WRAP" LINE 10: VE
  RIFY ""

```

LISTING 1: WORD WRAP UTILITY

For machine code buffs, here is the code disassembled and explained:

FDE8 EQU 65000

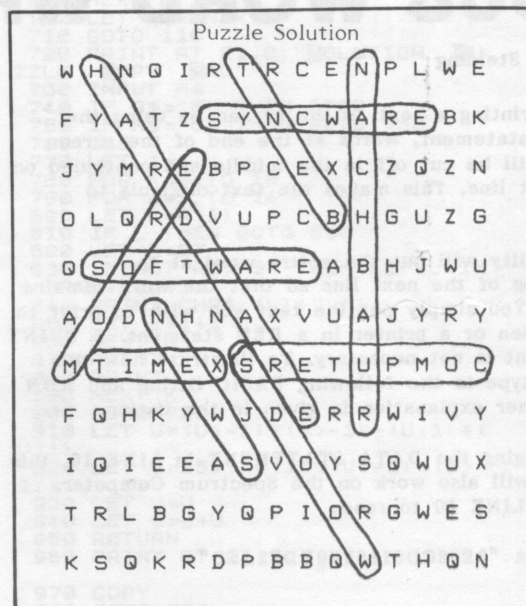
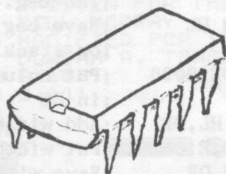
FDE8 3E02	wrap LD A,02	;Put 02 in the ;accumulator
FDEA CD3012	CALL chop	;Set to print to ;screen
FDED 3E0D	LD A,0D	;Put 0D in the ;accumulator
FDEF D7	RST 10H	;Do a carriage ;return
FDF0 2A5D5C	LD HL,(chad)	;Get addr. of REM ;statement
FDF3 23	INC HL	;Let HL=HL+1
FDF4 23	INC HL	;Let HL=HL+1 ;(HL=beg. of text)
FDF5 E5	CHEK PUSH HL	;Save beg. of line ;on stack
FDF6 112000	LD DE,0020	;Put column width ;in DE
FDF9 19	ADD HL,DE	;Add width to HL
FDEA 53	LD D,E	;Put width into D
FDFB D5	PUSH DE	;Save width on ;stack
FDFC 7E	LD A,(HL)	;What's at beg. of ;next line?
FDFD FE20	CP 20	;Is it a space?
FDFE 2015	JR NZ,LETR	;If not, goto LETR
FE01 C1	PRNT POP BC	;Get B counter off ;stack
FE02 E1	PRN1 POP HL	;Get text location ;off stack
FE03 7E	LOOP LD A,(HL)	;Put letter at HL ;in accum.
FE04 FE0D	CP 0D	;Is it a carriage ;return?
FE06 C8	RET Z	;If yes, return to ;Basic
FE07 D7	RST 10H	;If not, print it ;to screen
FE08 23	INC HL	;Let HL=HL+1 (get ;next letter)
FE09 10F8	DJNZ,LOOP	;If B<>0, goto LOOP
FE0B 3E0D	NEWL LD A,0D	;Load accum. with ;carriage ret.
FE0D D7	RST 10H	;Print it to screen
FE0E 7E	NEW1 LD A,(HL)	;Put letter at HL ;in accum.
FE0F FE20	CP 20	;Is it a space?
FE11 20E2	JR NZ,CHEK	;If not, goto CHEK
FE13 23	INC HL	;If yes, chk for ;another space
FE14 18F8	JR NEW1	;Goto NEW1
FE16 C1	LETR POP BC	;Get B counter off ;stack
FE17 2B	LTR1 DEC HL	;HL=HL-1 (find ;space)
FE18 05	DEC B	;B=B-1 (update ;counter)
FE19 7E	LD A,(HL)	;Put letter at HL ;in accum.
FE1A FE0D	CP 0D	;is it a carriage ;return?
FE1C 28E4	JR Z,PRN1	;If yes, goto PRN1
FE1E FE20	CP 20	;Is it a space?
FE20 20F5	JR NZ,LTR1	;If not, goto LTR1
FE22 18DE	END JR PRN1	;If yes, goto PRN1

'IMPROVED' from page 1

Other than its keyboard, one of the biggest complaints about the ZX81/TS1000 is its low resolution graphics. Those of us that have been struggling along with the graphics will be excited to read Wilf Rigger's article. In it he presents an honest-to-goodness software-only high-resolution screen display program for the ZX81/TS1000. A whole new world of screen displays awaits exploration.

Have any of you TS2068 users ever turned a little green with envy when looking at your buddies' QL or Mac-alike computers that could do pull down windows? You'll be back in the pink again after reading Paul Bingham's article. He has come up with an easy to program and use machine code window routine. Add this to your programs and be the envy of your buddies.

Get those 'new and improved' computers going! Don't forget the CONTEST!



SyncWare News

CONTENTS

For Your Support...	2
Forum...	3
2068 Label Maker...	7
Dertz & Swoger	
TS1000 Advanced Calculator	
Program...	9
Dean	
2068 Windows...	11
Bingham	
WRX16 HI-RES for TS1000...	13
Rigger	
2068 Variable SAVE Trick...	19
Fischer	
Basil's Compendium...	20
Wentworth	
TS1000 Wordsearch...	21
Teague	
2068 Word Wrap Utility...	23
Stelling	

FIRST-CLASS MAIL
U.S. POSTAGE
PAID
Jefferson, NH
Permit No. 1

Z.A. Lewis
PO Box 559
Camino, CA

95709

4:4

Editor:
Jeffrey Moore
602 S. Mill Street
Louisville, OH 44641

Technical Director:
Fred Nachbaur
C-12, Mtn. Stn. Group Box
Nelson, BC V1L 5P1 Canada

Advertising, Subscriptions:
Thomas B. Woods
P.O. Box 64
Jefferson, NH 03583

Submissions, announcements, letters to the editor, and all other material of editorial interest should be sent to the Ohio address.

Copyright 1986 SyncWare News

Subscribe ! \$16.95

For Canada and Mexico please add \$3.00 postage.
All other foreign add \$9.00 Airmail (U.S. funds only)

1 year (6 issues)